

2174

Delta Modulation

Final Report

January 1, 1973 - December 31, 1973

National Aeronautics and Space Administration

Johnson Space Center - Houston

under

NASA Grant NGR 33-013-063

(NASA-CR-134257) DELTA MODULATION Final  
Report, 1 Jan. - 31 Dec. 1973 (City  
Coll. of the City of New York.) 75 p  
HC \$6.75

CSCL 17B

G3/07

N74-21805

Unclas  
38210

COMMUNICATIONS SYSTEMS LABORATORY  
DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator



THE CITY COLLEGE OF  
THE CITY UNIVERSITY OF NEW YORK

Delta Modulation

Final Report

January 1, 1973 - December 31, 1973

National Aeronautics and Space Administration

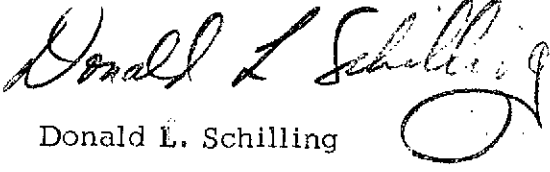
Johnson Space Center - Houston

under

NASA Grant NGR 33-013-063

COMMUNICATIONS SYSTEMS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING

A handwritten signature in dark ink, reading "Donald L. Schilling". The signature is fluid and cursive, with a large, stylized "S" at the end.

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator

Table of Contents

Introduction

I. An Overshoot Suppression Algorithm in Adaptive Delta Modulation

II. Rise Time, and Truncation Error

III. Settling Time

IV. A Technique for Correcting Transmission Errors in Delta Modulation Channels

V. Appendix

VI. New Areas of Research

VII. Ph. D. Dissertation

VIII. Papers Published and Presented

## Introduction

The final report summarizes several of the areas of research in Delta Modulation being supported under NASA Grant NGR 33-013-063 during the period January 1, 1973 - December 31, 1973.

1. The overshoot suppression algorithm described in the previous status report has been more extensively studied and investigated. An elaborate video simulation set-up has been developed using a PDP-8 computer. The computer-generated test-pictures showed a radical improvement due to the overshoot suppression algorithm.

Work is now underway to study the effects on real pictures from a flying spot

scanner. A paper describing the overshoot suppression algorithm has been presented at NTC - 73. A new paper<sup>\*</sup> which will include the overshoot suppression algorithm as well as the video simulation results is now being prepared to be submitted for publication in the IEEE Transactions on Communications.

2. Considering the Delta Modulator link as a nonlinear digital filter, a formula that relates the minimum rise time that can be handled for given filter parameters and voltage swings has been developed. In developing the formula the problem of random truncation errors due to the finite arithmetic implementation is handled in a deterministic manner with excellent results for all practical cases encountered. An approximate formula for the rise-time which neglects truncation error has also been developed. This formula, which gives reasonably good results, is far simpler than the exact formula.

3. The settling time, which is the time from the initial overshoot to the time at which steady state is reached, has been calculated for the case of overshoot

---

\* Sections I, II, and III of this report are essentially taken from this paper.

suppression as well as when no suppression is employed. The results indicate a significant decrease in settling time when overshoot suppression is used.

4. An Algorithm for correcting channel errors has been developed. It is shown that pulse stuffing PCM words in the DM bit stream results in a significant reduction in error-length.

The results presented in this report represent a significant step forward in the design of delta modulators for video encoding.

The reseach supported by this grant and its "sister" grant NGR 33-013-077 has resulted in 1 PhD Dissertation and the publication and presentation of several papers.

Participating in this program are:

Drs. I. Paz, W. Rosenberg and D. L. Schilling

and

Messrs. Z. Ali, E. Feria, J. LoCicero, M. Steckman, N. Scheinberg and  
L. Weiss

# I. Overshoot Suppression in Digital Adaptive Delta Modulation and Video Simulation Results

An overshoot suppression (O.S.S.) algorithm to improve the performance of the Digital Song Adaptive Delta Modulator (D.M.) for picture transmission is described. Figure 1 shows the structure of the digitally implemented optimum adaptive delta modulation system referred to in this report. Briefly, its operation is as follows:

The input signal  $S(t)$  is sampled and A to D converted to give  $S_k$ .  $S_k$  is then compared to its estimate,  $X_k$ , generating a sign-bit  $e_k$ , with

$$e_k = \text{sgn} (S_k - X_k) \quad (1)$$

where

$$X_k = X_{k-1} + \Delta_k \quad (2)$$

The step size at the  $k^{\text{th}}$  sampling instant is

$$\Delta_k = g_1 (e_{k-1}, \Delta_{k-1}) + g_2 (e_{k-2}, \Delta_{k-1}) \quad (3)$$

Thus the  $k^{\text{th}}$  step size depends on the previous step size, and the previous two sign bits. The  $g_1$  and  $g_2$  function characteristics are shown in Fig 2, indicating that:

$$\Delta_k = \begin{cases} |\Delta_{k-1}| (\alpha e_{k-1} + \beta e_{k-2}), & |\Delta_{k-1}| \geq 2 \Delta_o \\ 2 \Delta_o e_{k-1}, & |\Delta_{k-1}| < 2 \Delta_o \end{cases} \quad (4)$$

where  $\Delta_o$  is the minimum possible step size. The special region,  $|\Delta_{k-1}| < 2 \Delta_o$ , is needed to prevent a dead zone at the origin.

The decoder is just the feedback portion of the encoder. It reconstructs the approximation  $X_k$  from the  $e_k$  sign-bit stream.  $X_k$  is then D/A converted and low pass filtered to give  $\hat{S}(t)$ , the estimate of the transmitted signal.

In video processing,  $S(t)$  contains many large discontinuities of very short rise times. This corresponds to abrupt changes in grey level in the picture content. Thus, edge response is very important in video. A linear delta modulator is limited in its ability to track sudden input changes by its fixed step size, the magnitude of these steps being bounded by the permissible granular noise in the constant shade regions, see Fig 3a. Shade contrast is thus degraded by the so called "slope-overload-noise" introduced by the D. M. channel. To alleviate this condition and permit  $\hat{S}(t)$  to approximate rapid rises (i.e., minimize the slope-overload-noise) it is desirable to make the step sizes  $\Delta_k$  small in steady state but allow them to increase quickly, in some nonlinear fashion, when tracking rapidly varying inputs. This is done in an adaptive D. M. by adjusting  $\alpha$ ,  $\beta$ , and  $\Delta_o$  to meet rise time requirements. See Fig 3b and equations (1) through (4). Other algorithms are also being considered.

The sharp rises in a video signal are usually followed by regions of fairly constant level due to regions of uniform shade in the picture. Thus while alleviating slope overload problems, an adaptive D. M. introduces the possibility of large overshoots when the tracked level is finally reached. Furthermore, the overshoot is followed by a transient oscillatory response until the D. M. finally locks onto the tracked signal level. These effects are shown in Fig 3b. Figure 3b also shows that good steady state response, i.e., small amplitude oscillations about a constant level in  $\hat{S}(t)$ , requires a small minimum step,  $\Delta_o$ . It can also be shown that the D. M. becomes unstable if  $\alpha$  and  $\beta$  are made too large (see Appendix A). Thus in choosing  $\alpha$ ,  $\beta$ , and  $\Delta_o$  a trade-off must be made between slope overload noise versus overshoot, including the recovery (settling) time, and the requirements for steady state response. Moreover, all this must be done while maintaining D. M. stability.

We may therefore conclude as follows: both the overshoot and the subsequent recovery time are undesirable attributes of an adaptive D. M. Reducing the step

size decreases the possible overshoot amplitudes and shortens the recovery time. This, however, augments slope overload and rise-time. A trade-off therefore exists between overshoot amplitude and its recovery time, versus slope-overload-noise in an adaptive D. M.

Overshoot suppression is a scheme to sharply limit the overshoot amplitude and reduce the subsequent recovery time. This is done without reducing the step sizes until overshoot is imminent. Thus, slope overload noise, and hence the rise time, is decreased. Using this technique  $\alpha$  and  $\beta$  can be increased while decreasing  $\Delta_o$ . In this way, slope overload as well as small steady state response requirements can be met simultaneously. Impending instabilities due to large  $\alpha$  and  $\beta$  are also inhibited and, obviously, overshoot and the subsequent recovery time are minimized.

An O. S. S. algorithm has recently been suggested in the literature (3). This scheme uses a "look-up table" in which arbitrary values for the step sizes are suggested. Furthermore, the maximum step size is limited by overshoot considerations.

In the Song Delta Modulator, the step sizes can continually increase with the O. S. S. scheme described in this report, thus yielding better signal to slope-overload-noise ratio than that obtained using other O. S. S. techniques. Moreover, the amount of equipment involved in implementing our O. S. S. algorithm is very modest in comparison to the equipment needed to implement other proposed schemes, and could fit into any adaptive D. M. in which the next step size is explicitly calculated. It is also flexible as to the amount of O. S. S. it can perform and trade-offs between conflicting factors can be accurately set as may be necessary.

Our O. S. S. algorithm has been verified. The results obtained indicate that the rise time of the DM can be decreased without causing excessive overshoots and subsequent oscillations. The marked improvement in the quality of the transmitted video is illustrated with pictures processed by the D. M. with and without O. S. S.

#### A. The Proposed Overshoot Suppression Algorithm

The Overshoot Suppression Algorithm may be understood by considering the four cases shown in Fig 4 in which an overshoot or an undershoot occurs. In Fig 4a an overshoot occurs at sampling time  $k-1$  followed immediately by an

undershoot at  $k$ . For this case it is easy to show that the D. M. considered will approach its steady-state condition rapidly. This is not the case in Fig 4b where the overshoot is larger than in (a) and  $X_k$  is greater than  $S_k$ . Consequently an undershoot occurs at  $k + 1$  or later and with an amplitude larger than in (a). This occurs because the step sizes begin increasing again after the first reversed step. Thus it will take many more sampling periods to reach steady state in (b) than in (a). The algorithm is therefore implemented only when case (b) occurs. Note that Figs 4(c) and 4(d) depict undershoots corresponding to the overshoots in (a) and (b) respectively. Action to prevent excessive undershoots is thus taken only for case (d).

The occurrence of cases (b) and (d) can be recognized by examining the sequence  $(e_{k-3}, e_{k-2}, e_{k-1}, e_k)$ . The fingerprint of (b) is  $(+1, +1, -1, -1)$ , while that of (d) is  $(-1, -1, +1, +1)$ . When either sequence is encountered action is taken to prevent overshoot or undershoot. The corrective action entails decreasing the stored values of  $\Delta_{k-1}$ , and hence  $X_{k-1}$ , and  $X_k$ . Case (b) is thus transformed into a case (a) situation and the same for (d) and (c) respectively. The shape of the modified waveform actually depends on the amount by which  $\Delta_{k-1}$  is decreased. The simplest scheme is one where  $\Delta_{k-1}$  is replaced by half of its original value. We may allow for more rapidly increasing step sizes,  $\Delta_k$ , i.e., larger  $\alpha$  and  $\beta$  (see Eq 4), as long as  $\Delta_{k-1}$  is replaced by a smaller fraction of its original value when O. S. S. is employed. Thus there is a faster initial rise coupled with a very sharp braking action just before the desired level is reached. However, since the braking action occurs close to the desired level there is nominal slope overload degradation. Indeed, there is an overall decrease in slope overload noise due to the more rapid increase in the initial step sizes.

The Overshoot Suppression Algorithm was applied to the adaptive D. M. operating in the Song Mode, i.e.,  $\alpha = 1, \beta = 0.5$ . It was shown (3) that a usable video system results using these parameters even without O. S. S. With the addition of the suppression algorithm video reproduction is significantly improved.

The salient features of the Song Mode response are now summarized: In approaching a level from above or below as in Fig 4, each step size is 1.5 times the previous one (see Eq 4 for  $\alpha = 1, \beta = 0.5$ ). When a direction reversal occurs,

as at sampling time  $k$  in Fig 4, the first step size following the reversal is one half the previous step size, i.e.,  $\Delta_k = -1/2 \Delta_{k-1}$  (see Eq 4). Thus, in Fig 4(b) we have

$$X_k \geq X_{k-1} - (1/2)\Delta_{k-1} = X_{k-2} + (1/2)\Delta_{k-1} \quad (5)$$

The inequality sign is needed due to the fixed point arithmetic employed in the digital implementation. Also in Fig 4(b),  $X_{k-2} < S_k < X_k$ . To implement O. S. S., set  $(\Delta_{k-1})' = (1/2)\Delta_{k-1}$ , where the prime refers to the new values after the O. S. S. algorithm has been implemented. Therefore

$$(X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + (1/2)\Delta_{k-1} \quad (6)$$

Next, set

$$(\Delta_k)' = \Delta_k = -(1/2)\Delta_{k-1} \quad (7)$$

Thus

$$(X_k)' = (X_{k-1})' + (\Delta_k)' = X_{k-2} \quad (8)$$

Hence, Fig 4(b) has been transformed into Fig 5, with undershoot occurring at  $k$  rather than at  $k+1$  or later. It should be evident from Fig 5 even without a detailed explanation of the worst case that the overshoot has been at best entirely eliminated or at worst cut in half depending on whether  $S(t)$  is closer to  $(X_{k-1})'$  or  $X_{k-2}$  respectively. Figure 5 also shows that the recovery time is greatly reduced, since the D. M. locks onto  $S(t)$  very rapidly after sampling time  $k$ . Note also that now  $(e_k)' = \text{sgn}(S_k - (X_k)') = 1$ , whereas in Fig 4(b)  $e_k = \text{sgn}(S_k - X_k) = -1$ .

The above O. S. S. scheme is summarized in the form of an algorithm by considering a typical cycle of the now modified D. M.:

Step 1: Generate  $S_k$

Step 2: Calculate  $\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1})$

Step 3: Calculate  $X_k = X_{k-1} + \Delta_k$

Step 4: Calculate  $e_k = \text{sgn}(S_k - X_k)$  and transmit this bit.

In the D. M. without O. S. S. this would complete the cycle. That is,  $k$  is next updated and steps 1 through 4 are repeated. To implement O. S. S. the following additional steps are needed:

Step 5: If  $e_{k-3} = e_{k-2} = 1$ , and  $e_{k-1} = e_k = -1$ , set  $V = 1$ . If  $e_{k-3} = e_{k-2} = -1$ , and  $e_{k-1} = e_k = 1$ , set  $W = 1$ .

Step 6: If  $V \neq 1$  and  $W \neq 1$  go to step 7; If  $V = 1$  or  $W = 1$  set

$$(a) (\Delta_{k-1})' = (1/2) \Delta_{k-1}$$

$$(b) (X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + (1/2) \Delta_{k-1}$$

$$(c) (\Delta_k)' = \Delta_k = -(1/2) \Delta_{k-1}$$

$$(d) (X_k)' = (X_{k-1})' + (\Delta_k)' = X_{k-2}$$

$$(e) (e_k)' = -e_k$$

Step 7: Update  $k$ . That is, set  $e_{k-3} = e_{k-2}$ ;  $e_{k-2} = e_{k-1}$ . If step 6 has been executed set  $e_{k-1} = (e_k)'$ ;  $X_{k-2} = (X_{k-1})'$ , etc. Otherwise set  $e_{k-1} = e_k$ ;  $X_{k-2} = X_{k-1}$ , etc. This completes a cycle.

## B. Hardware Implementation of the Overshoot Suppression Algorithm

The implementation of the above O. S. S. algorithm requires the addition of very little hardware to the Digital Song Adaptive Delta Modulator. This can be seen by considering the schematic representation of the D. M. CODEC (Coder-Decoder-Combination) with O. S. S. shown in Fig 1. Note that the extra components needed to implement the suppression scheme appear in branches that are drawn with dashed lines. Of these, the only major devices are the delay elements D5, D6, and the adder A4. However, since the adders A1, A2, and A3 are really one time-shared adder, we can easily time-share adder A4 also. The remaining extra elements are a few gates needed for decision, switching and

timing purposes. Note that the execution of step 6(a) of the algorithm,  $(\Delta_{k-1})' = (1/2) \Delta_{k-1}$ , need not require the addition of any explicit hardware. We merely read into adder A4 the contents of the  $\Delta_{k-1}$  register shifted by one bit, thereby resulting in a division by two.

It is difficult to discern the operation of the circuit by merely examining the schematic diagram in Fig 1 because the sequential order of operation is not specified in the diagram. However, the actual operation is made clear by considering Fig 1 in conjunction with the seven steps of the O. S. S. algorithm.

The additional steps of the algorithm place an added requirement on the logic speed. After the completion of a normal cycle of the D. M., extra time is needed to perform one more addition and the various logic operations needed to rearrange the internal values.

At this point, it should be pointed out that a significant simplification is possible in the encoder implementation. Namely, Step 6(b) and hence Step 6(a) do not have to be explicitly executed in the encoder because  $(X_{k-1})'$  is not really needed to compute  $(X_k)'$  in Step 6(d), i.e.,  $(X_k)'$  is simply replaced by  $X_{k-2}$  which is available in the memory. Furthermore, it is easy to show that  $(X_{k-1})'$  is not used in later cycles due to the fact that once an overshoot is suppressed at  $k-1$ , the earliest future time for implementing the algorithm is at  $k+2$ . By this time  $X_{k-1}$  is clocked out of the memory. In terms of hardware savings in the encoder, this eliminates the gates needed to produce  $(\Delta_{k-1})' = (1/2) \Delta_{k-1}$ , as well as A4 and A2 to carry out Steps 6(b) and 6(d) respectively. These simplifications are not possible in the decoder because its output with O. S. S. has to be taken from  $X_{k-1}$  rather than from  $X_k$ . Note that if the output was taken from  $X_k$ , then the O. S. S. produced by going back in time and reducing  $X_{k-1}$  would not be evident in the output  $\hat{S}(t)$ .

### C. Computer Verifications of the Algorithm

The Digital Song Delta Modulator, with and without O. S. S. was simulated on a PDP-8 computer. The minimum step size used ( $\Delta_0$ ) was normalized to unity. The dynamic range was set to correspond to a ten bit internal arithmetic

in an actual hardware implementation. Thus, the step size was permitted to vary from  $\Delta_o$  to  $1024 \Delta_o$ .

The responses of the D. M. to step functions of different amplitudes, with and without O. S. S., appear in Fig 6. Figures 6(a), and 6(b) exhibit large overshoots and sustained oscillations (see also Fig 9(b)). They correspond to the sequence  $e_{k-3} = e_{k-2} = 1$ ,  $e_{k-1} = e_k = -1$ , where  $k-1$  is the sampling time when overshoot occurs. Figures 6(a') and 6(b') are the same waveforms but with overshoot suppression (see also Fig 9(c)). As an example compare Figs 6(a) and 6(a'). Here the maximum peak to peak oscillations are reduced from  $22\Delta_o$  to  $9\Delta_o$ . Similar observations can be made for Figs 6(b) and 6(b'). Furthermore, here the settling time to the steady state is reduced from six to three sampling periods. While Fig 6 gives a good indication of the general nature of the improvement due to O. S. S., a more convincing illustration is depicted in Fig 7 where the discontinuities are much larger. Note that the apparent slow rise times in Fig 7 are due to the compression produced by a scaling factor of 0.1 used in the plotting. In reality Fig 7 rises over a range of  $500 \Delta_o$  in only 13 sampling periods. To achieve the same amplitude, a non adaptive D. M. would require 500 sampling periods.

Briefly, the salient features of the response are as follows: The rise time to reach a given level is the same with or without O. S. S. Overshoots are suppressed by a minimum of 50%. Recovery times following overshoots are significantly reduced as seen in Fig 7(b). The data plotted in Fig 7 is given in Table 1 for quantitative comparison. The peak-to-peak amplitude of the steady state response is three times the minimum step size for either scheme. The period of steady state oscillations is 4 sampling periods without O. S. S., and 8 sampling periods with O. S. S. In either case, the peak-to-peak steady state oscillation amplitudes are smaller than a grey level in the picture waveform. Thus constant shade regions will not suffer significant degradation (granularity and contouring).

#### D. Experimental Investigations

##### 1. The Test Set Up

3 provides the necessary staircase waveform for its vertical deflection system. For steady pictures proper synchronization between the horizontal and the vertical waveforms has been provided.

The final result of applying the outputs of Channels 1, 2, and 3 to the Z, horizontal trigger, and vertical inputs, respectively, of the display monitor is an artificial still picture. The picture contains N bright vertical bars formed by M vertically displaced raster lines, N and M being externally specified. The effects of the DM edge response as well as the steady state granular noise on a video display are very prominent in this resulting bar pattern.

The display module turns out to be the fastest of the set. As a result, the repetition rate of the display waveform is 3.5 times that of the original waveform generator output. This fact was accounted for in the scaling factors that govern the bandwidths of the L. P. filters.

## 2. Experimental Results

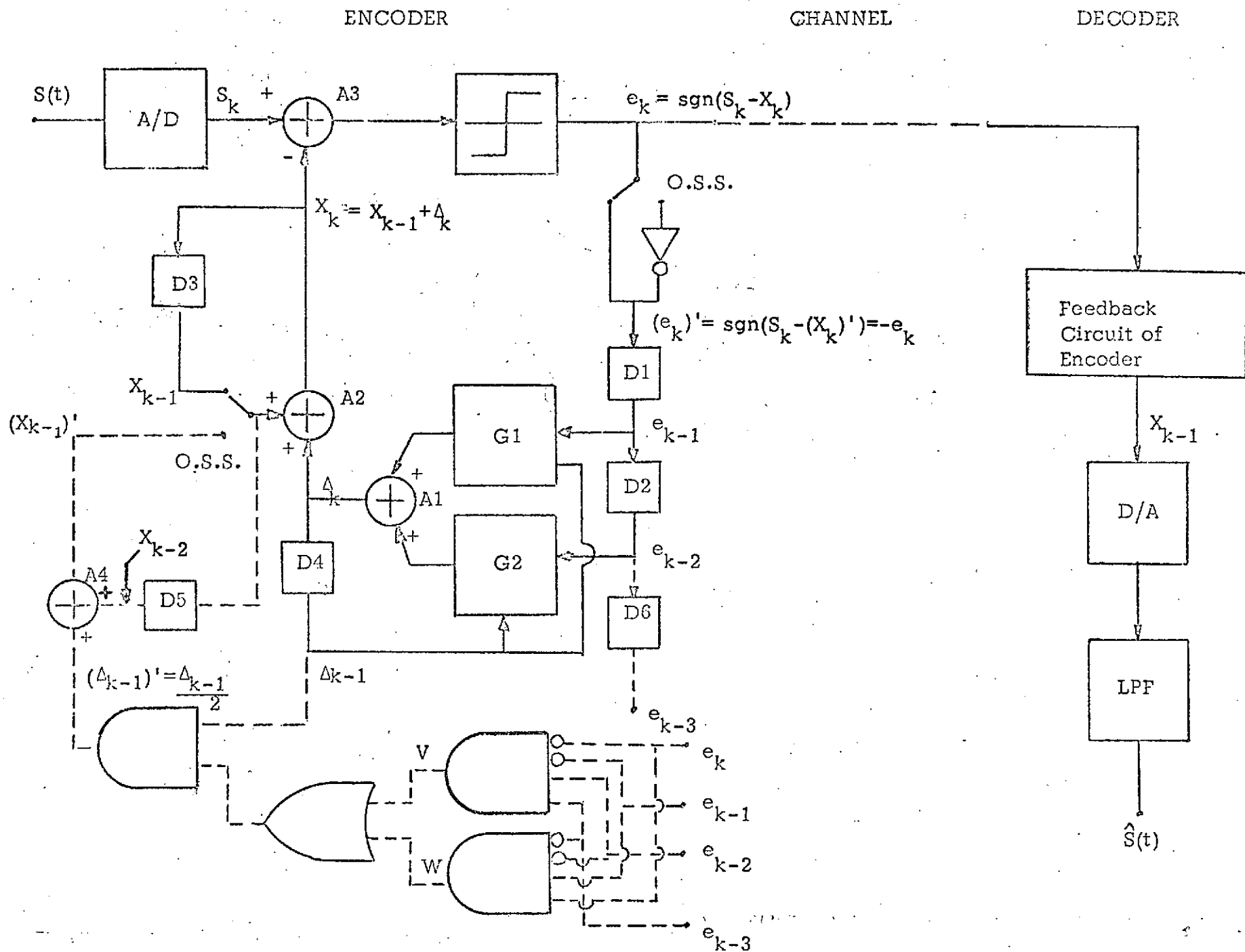
The experimental results are summarized in the photographs appearing in Fig 9. The waveform employed was a 5V peak-to-peak square wave having a repetition frequency of 100Hz and sampled at a frequency of 11.4 KHz, i.e., 114 samples/period. The square wave is bandlimited to 3KHz. In the final waveforms displayed in the photographs the time is scaled by a factor of 3.5 by the display program. That is, the repetition rate is 350 Hz, the sampling frequency is 39.9 KHz, and the bandwidth is 10.5 KHz.

The original square wave and the filtered waveforms appear in Fig 9(a). Fig 9(a') is the 100 line raster produced with the filtered waveform of Fig 9(a) as the Z modulation. Note that except for the narrow vertical bright band at the leading edge of each bar, which is due to the ringing introduced by the filter, the brightness of the bar is uniform and the edges are sharply defined. The bright (or dark) band that cuts horizontally along all vertical bars is simply due to a lack of perfect synchronization between the camera and the vertical sweep.

Fig 9(b) shows the delta modulated version, without O. S. S., of the filtered waveform in Fig 9(a). The lower waveform in Fig 9(b) is just the delta

modulated output waveform bandlimited to  $f_m = 10.5 \text{ KHz}$ , (the bandwidth of the input to the D. M.). The raster in Fig 9(b') results when the filtered waveform of Fig 9(b) is applied to the Z-input of the monitor. The deleterious effects of the overshoots and subsequent oscillations on the leading edge are obvious. Note also that the gradual roll-off at the trailing edge of the waveform causes a lack of sharp contrast at the trailing edge of the bright bars.

Figures 9(c) and 9(c') correspond to Figures 9(b) and 9(b') respectively, when O. S. S. is applied. The improvement in Fig 9(c') over 9(b') is obvious. In fact, the only difference between Fig 9(a') and Fig 9(c') is the lack of sharp contrast on the trailing edges in Fig 9(c'). This is due to the slower fall-time of the delta modulated waveform than that of the original filtered waveform. The way to improve on this is to increase  $\alpha + \beta$  which will lead to faster rise time capabilities of the D. M. This is now possible because the overshoot effects on the leading edge are taken care of by the O. S. S. scheme.



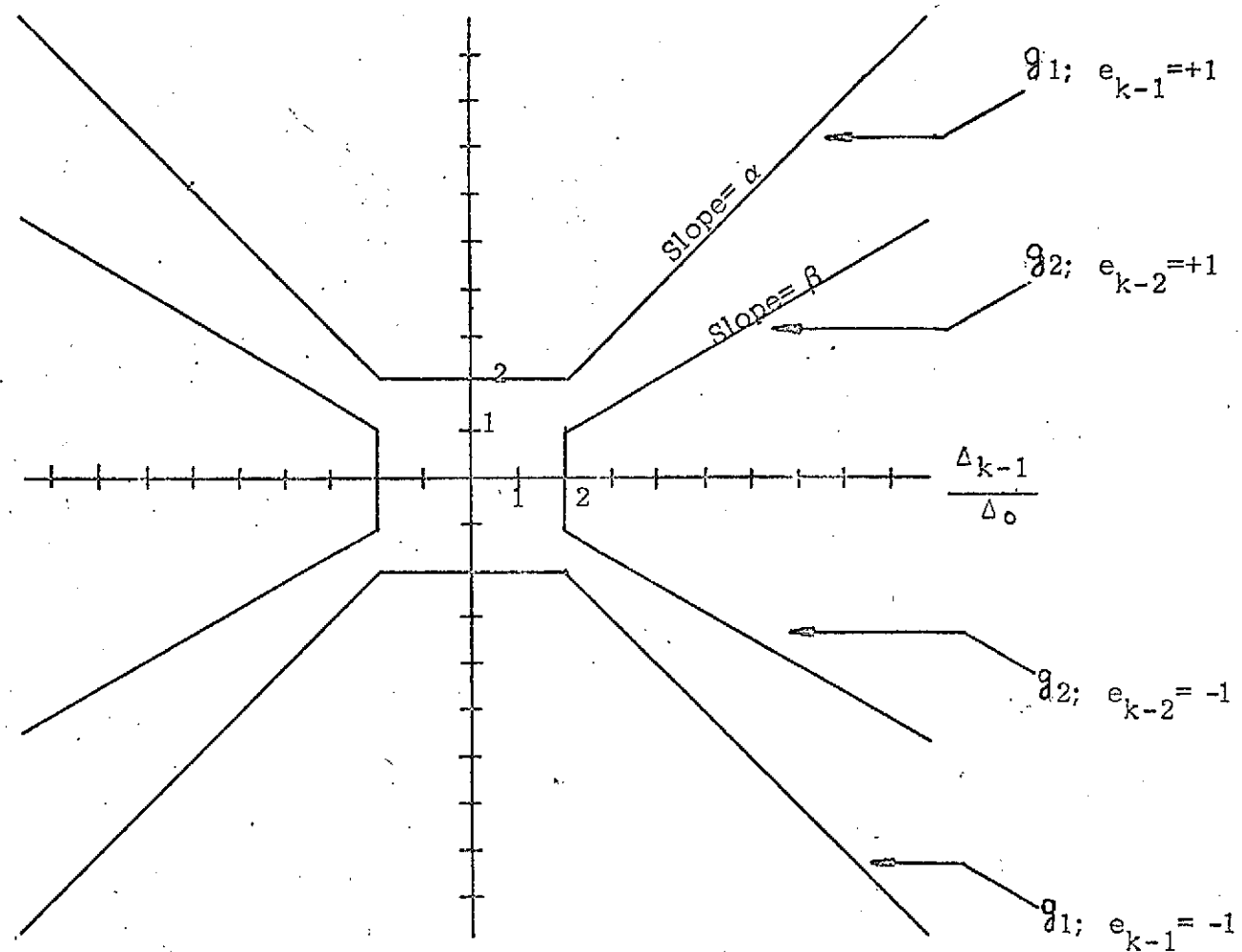


Fig. 2. Normalized  $g_1, g_2$  Characteristics. ( $\Delta_0$  = Minimum Step Size.)

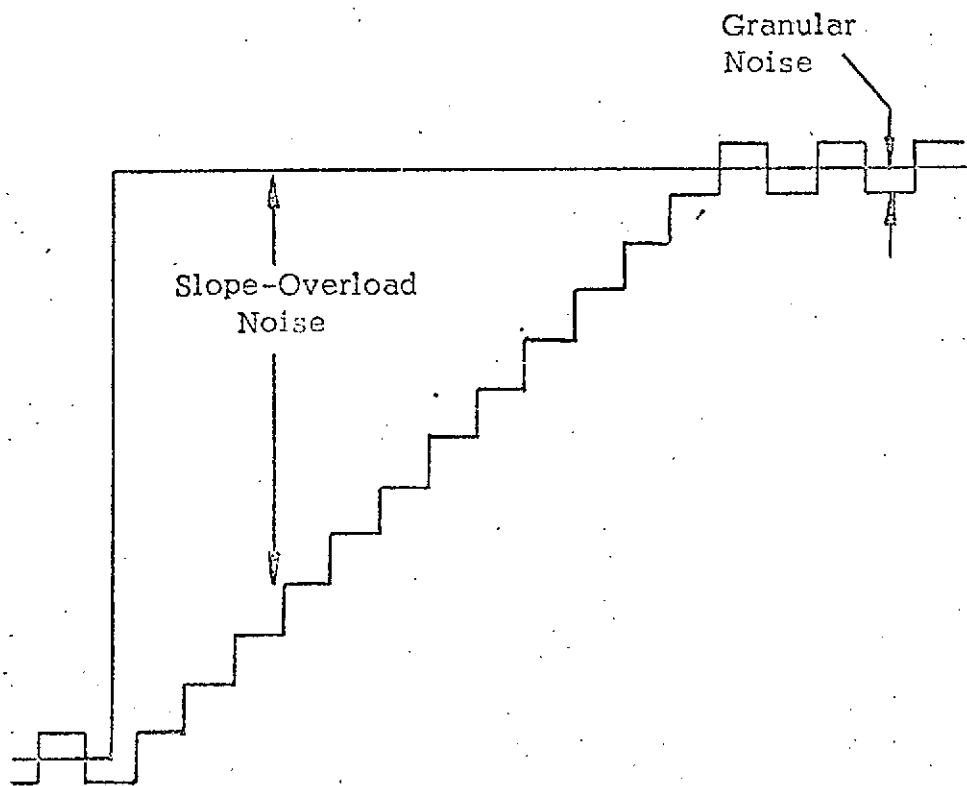


Fig. 3 (a) Linear Delta Modulator step response exhibiting Slope-Overload-Noise and Granular Noise.

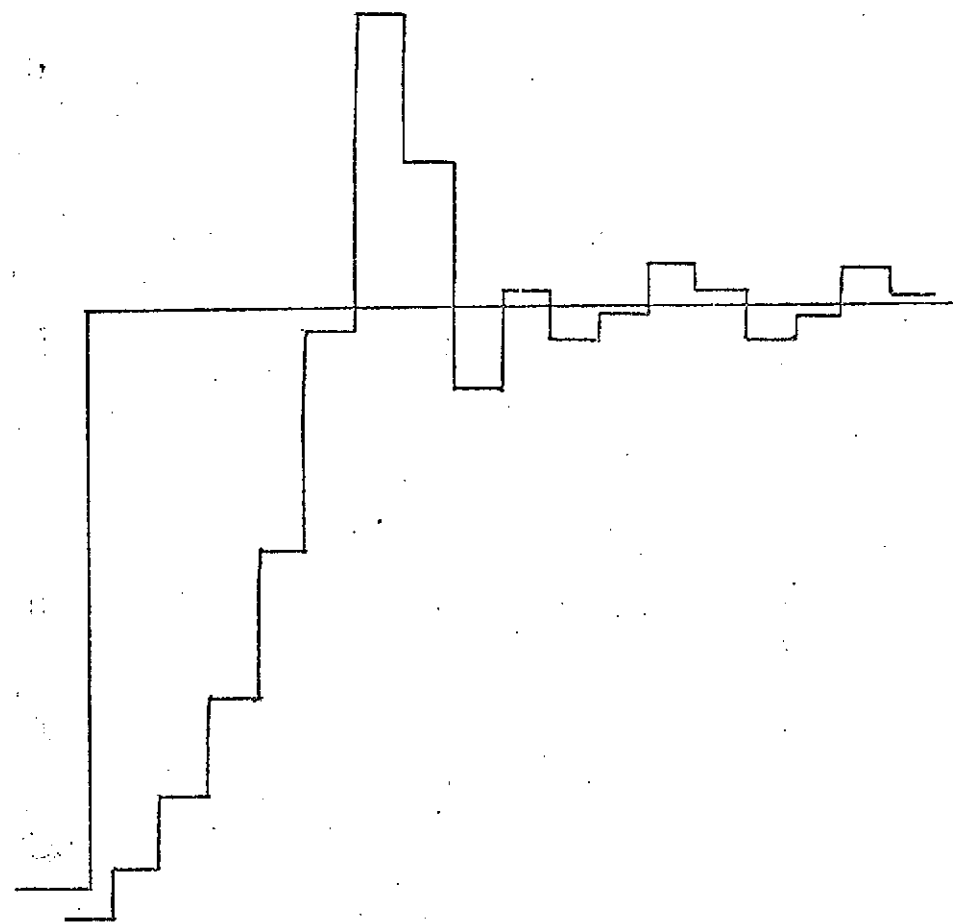


Fig. 3 (b) Adaptive Delta Modulator step response exhibiting overshoots and oscillations.

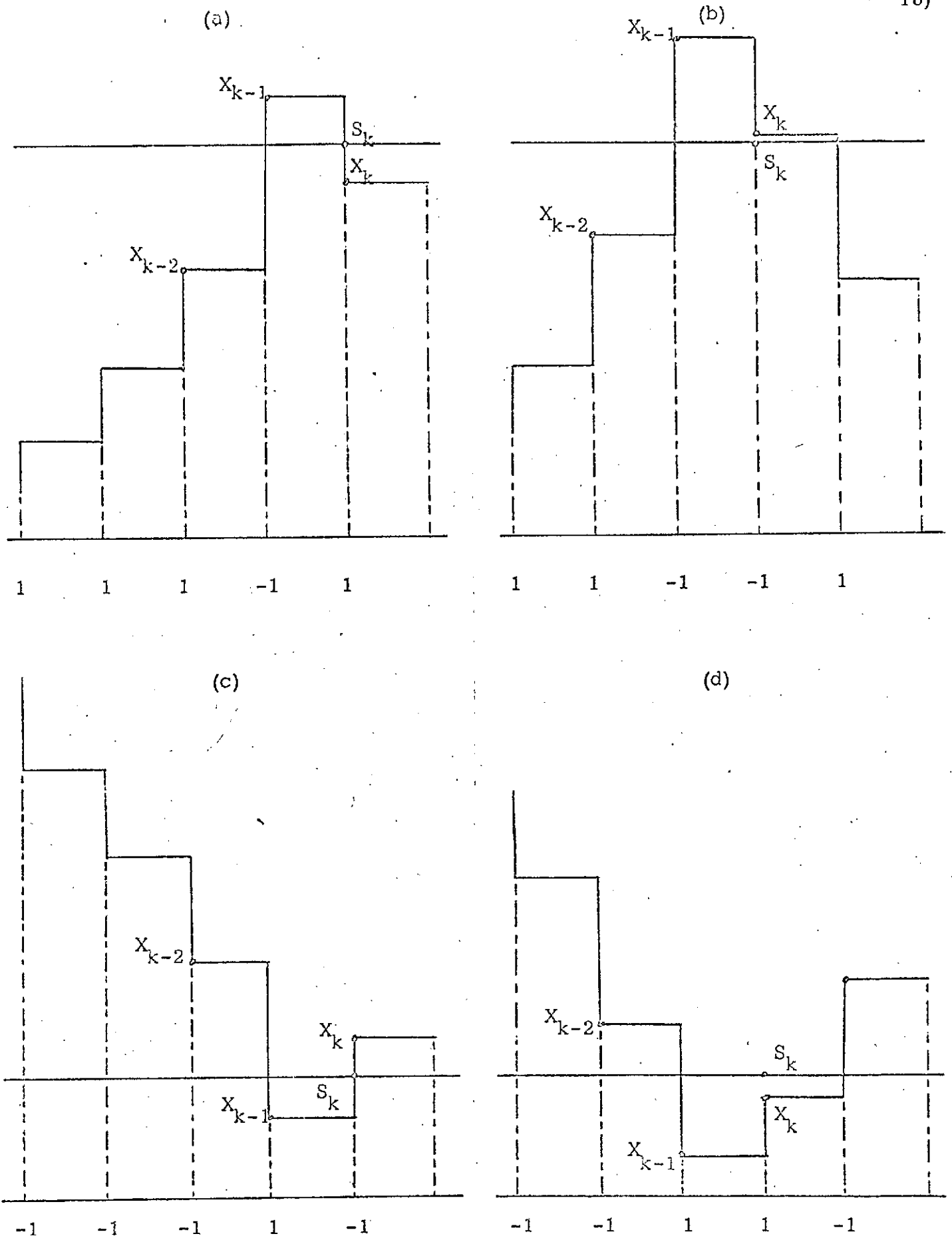


Fig. 4 The four cases of overshoot in the step response of the Song-Adaptive-Delta Modulator.

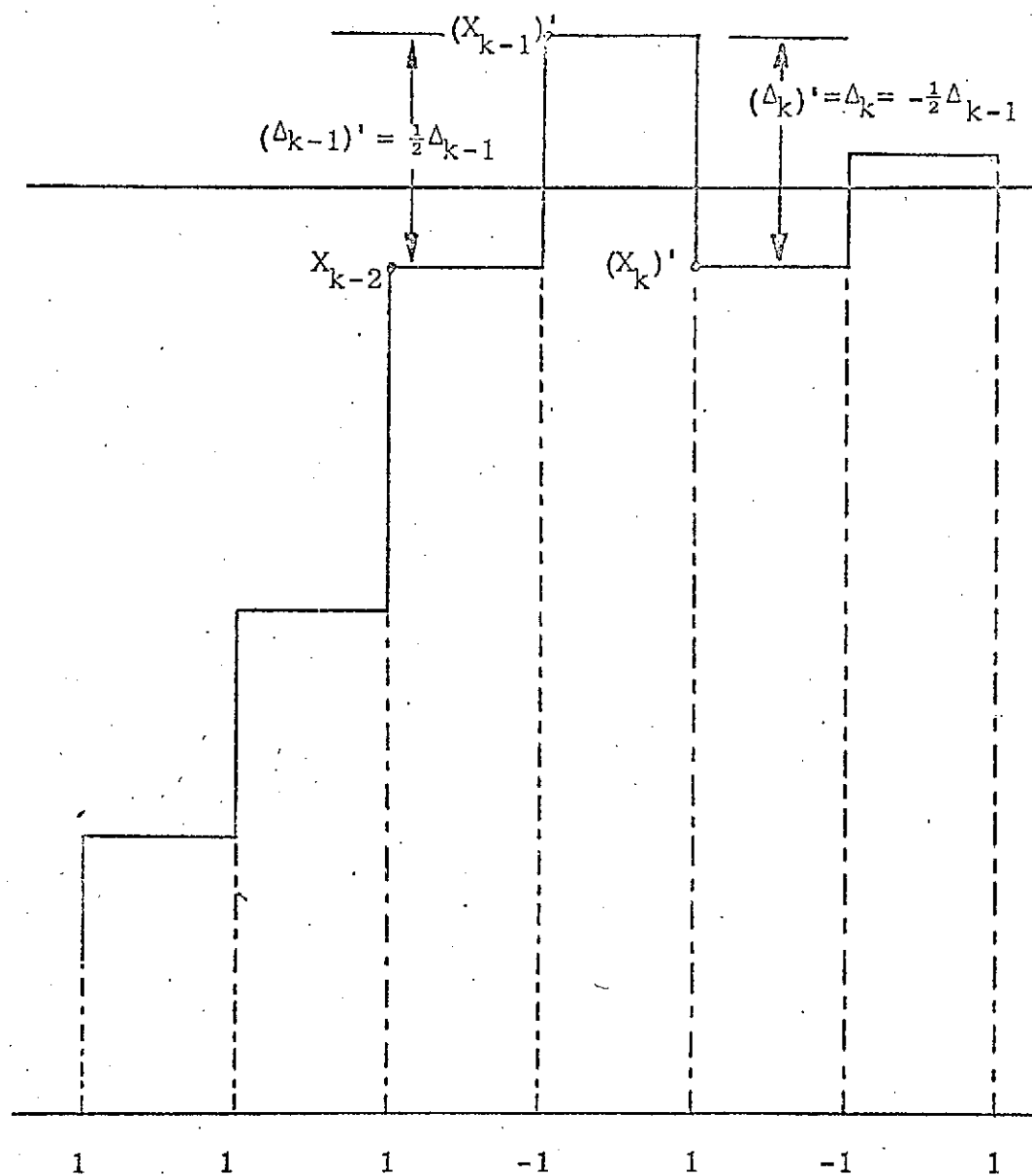


Fig. 5 Figure 4b after Overshoot Suppression.

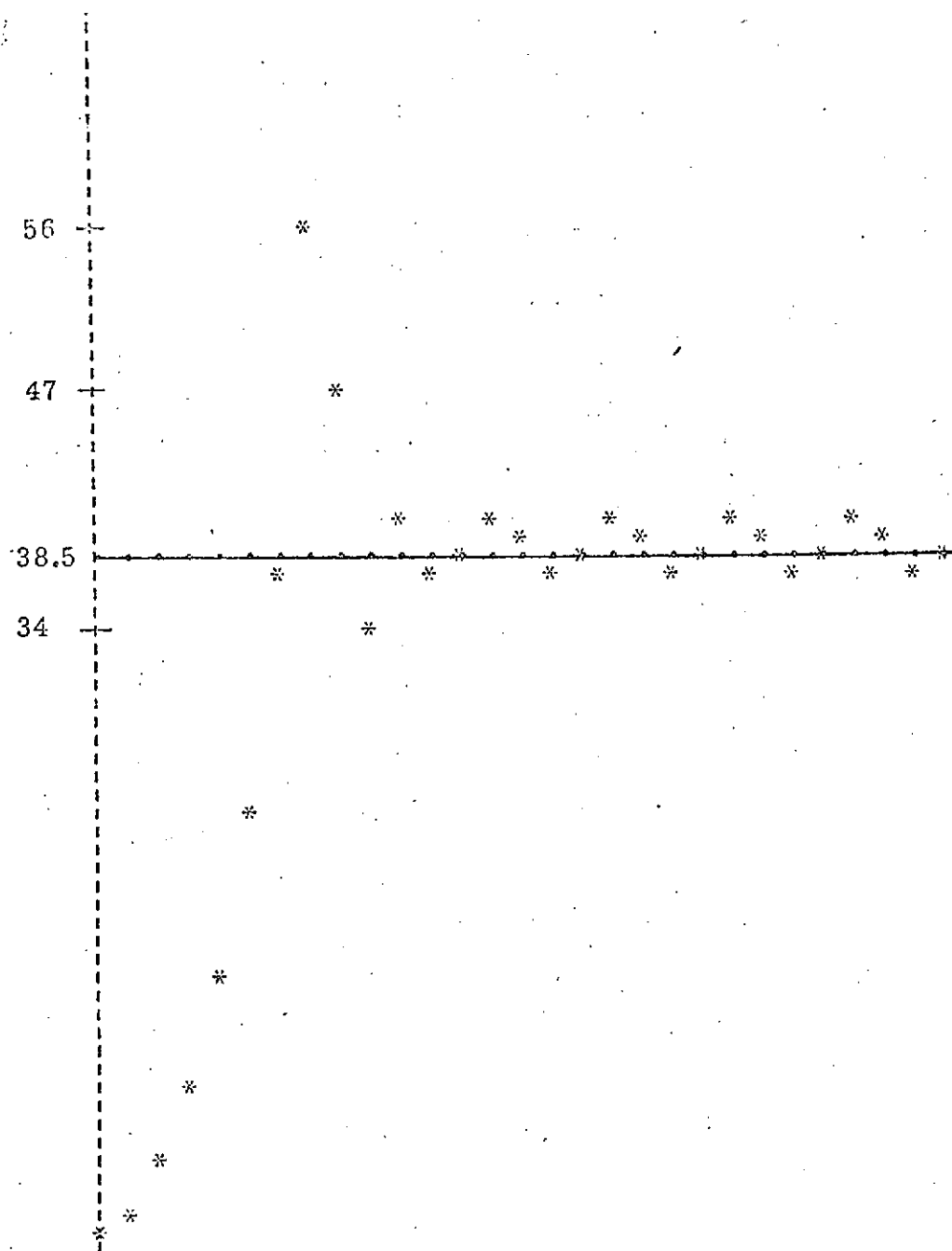


Fig. 6 (a) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

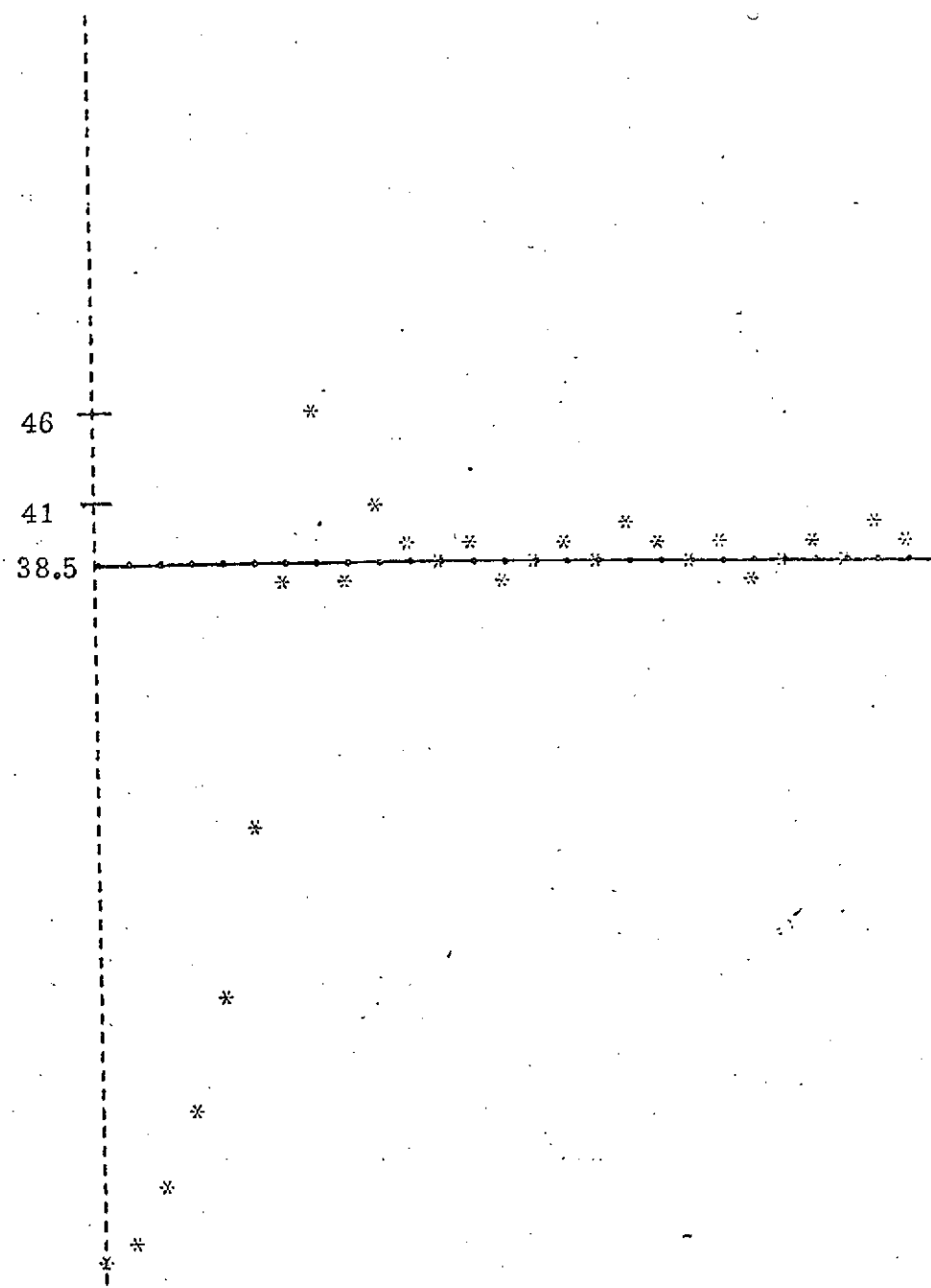


Fig. 6(a') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

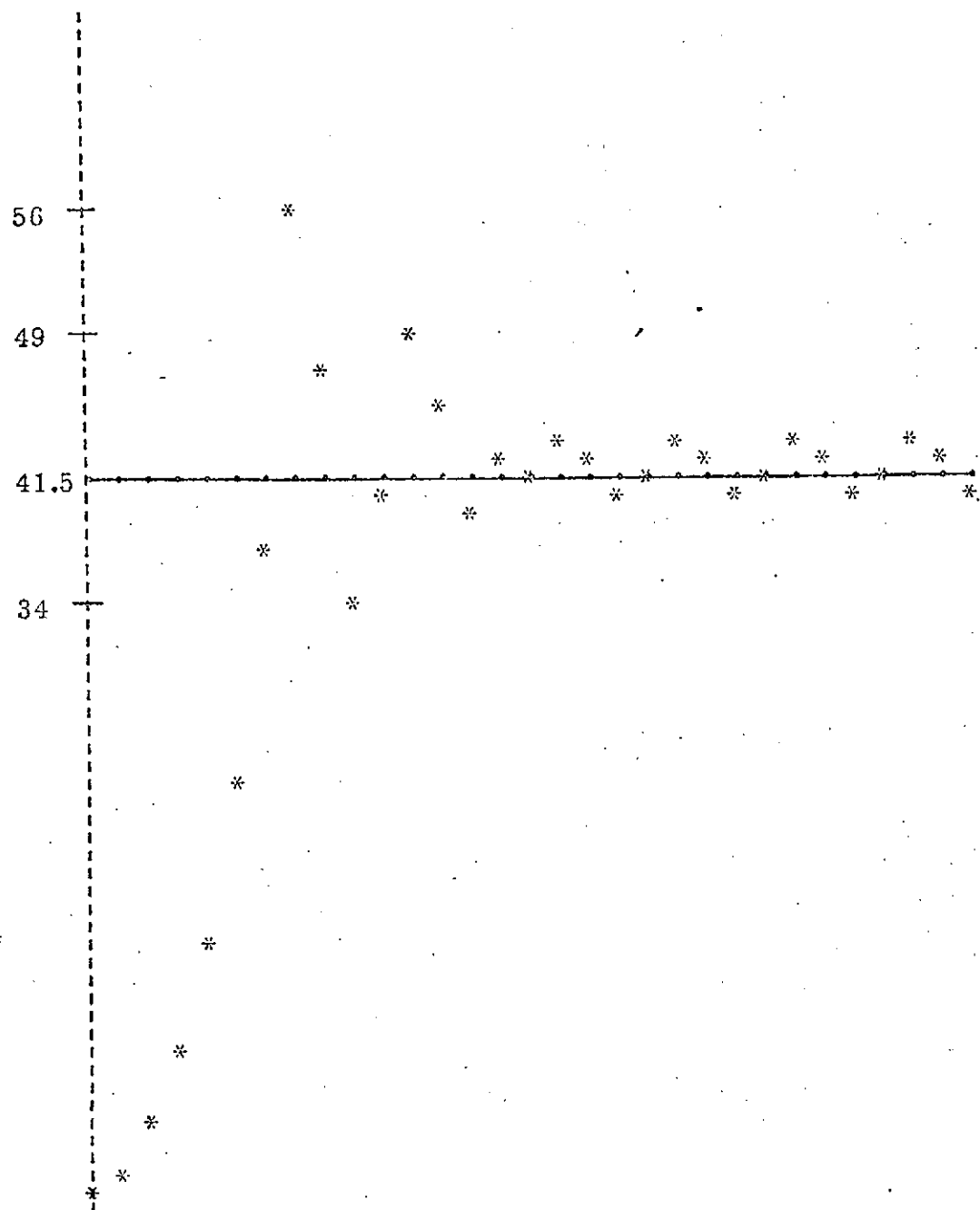


Fig. 6(b) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

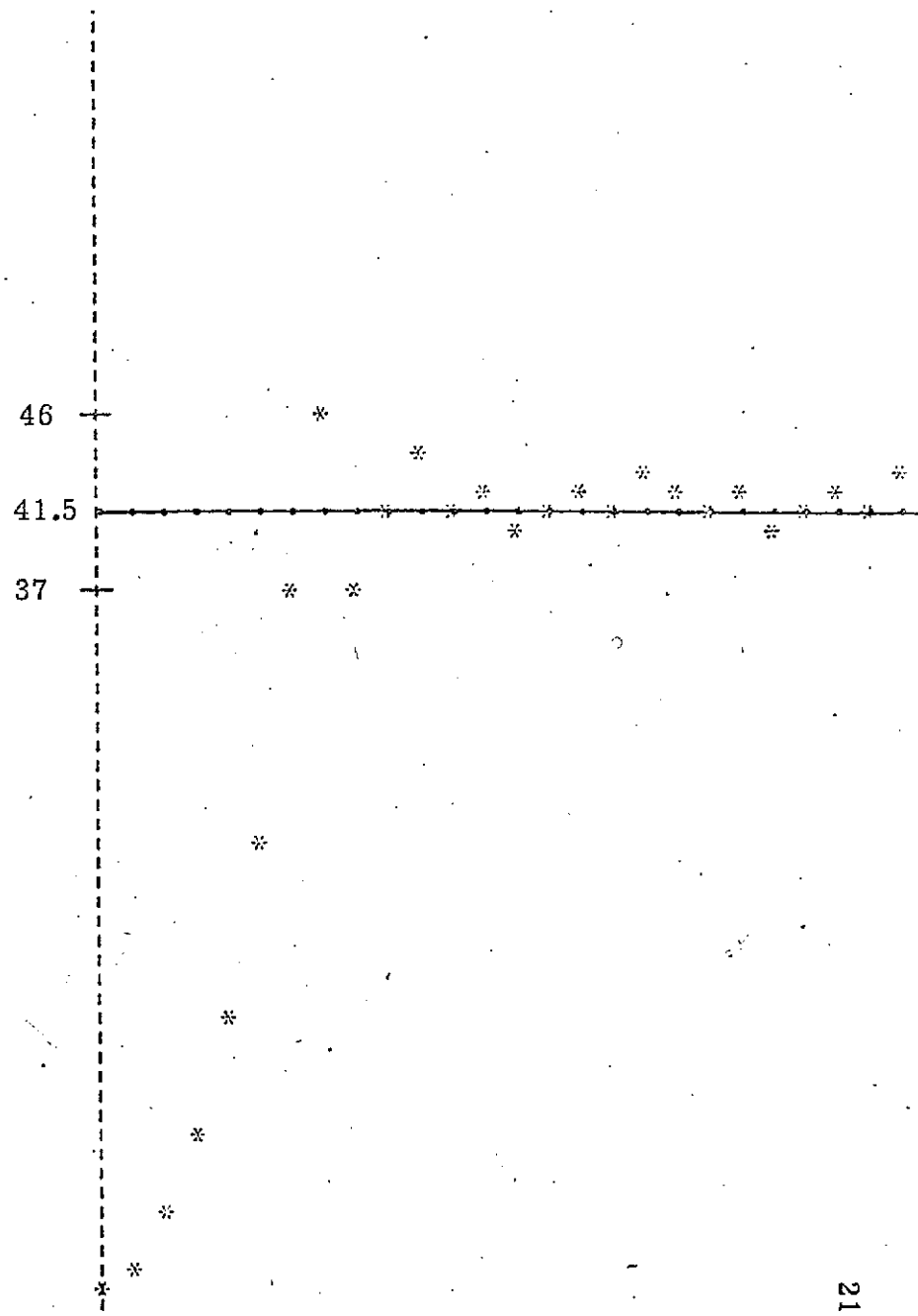


Fig. 6(b') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

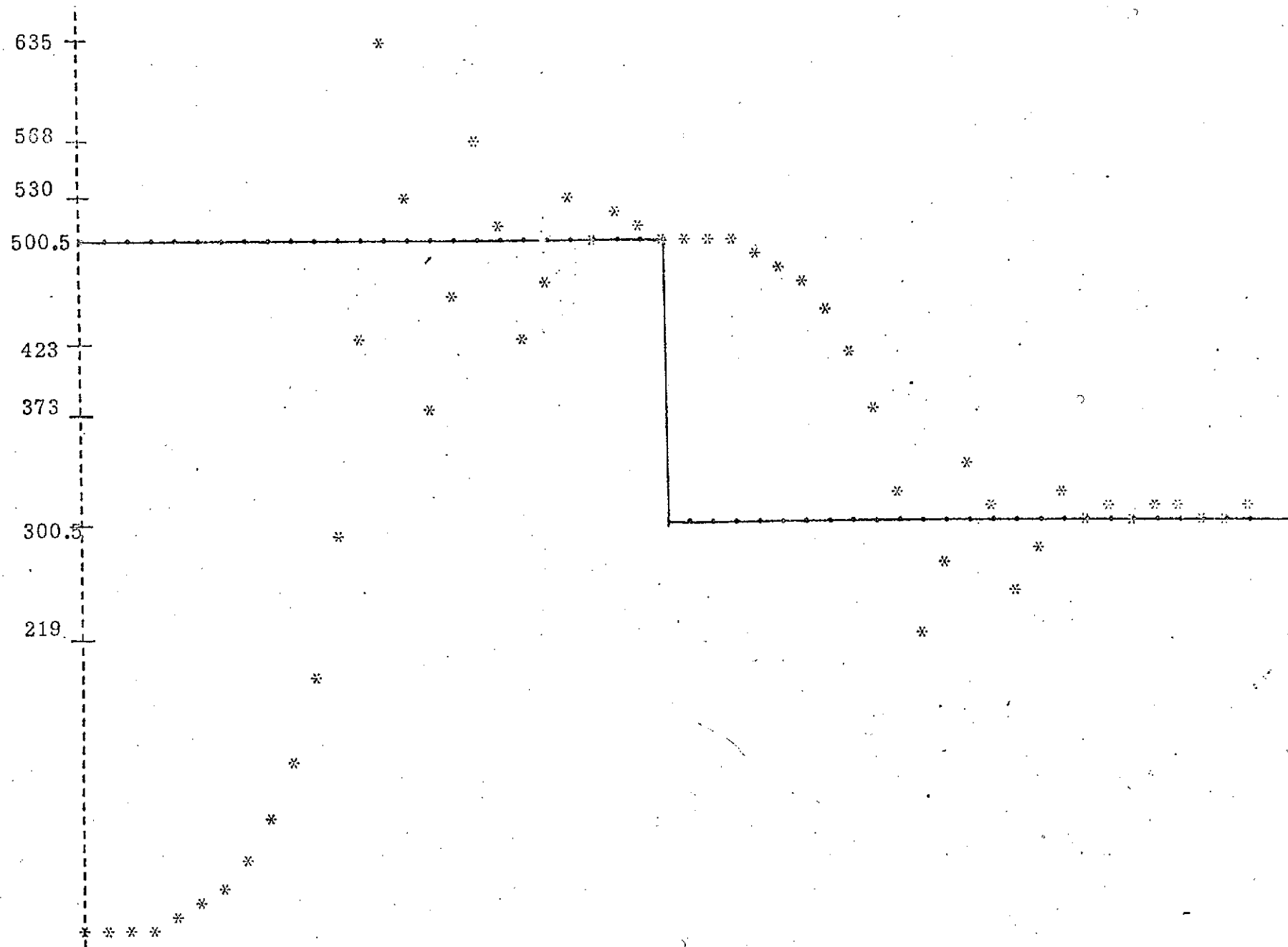


Fig. 7(a) The response of a Song-Adaptive Delta Modulator to abrupt level changes without Overshoot Suppression.

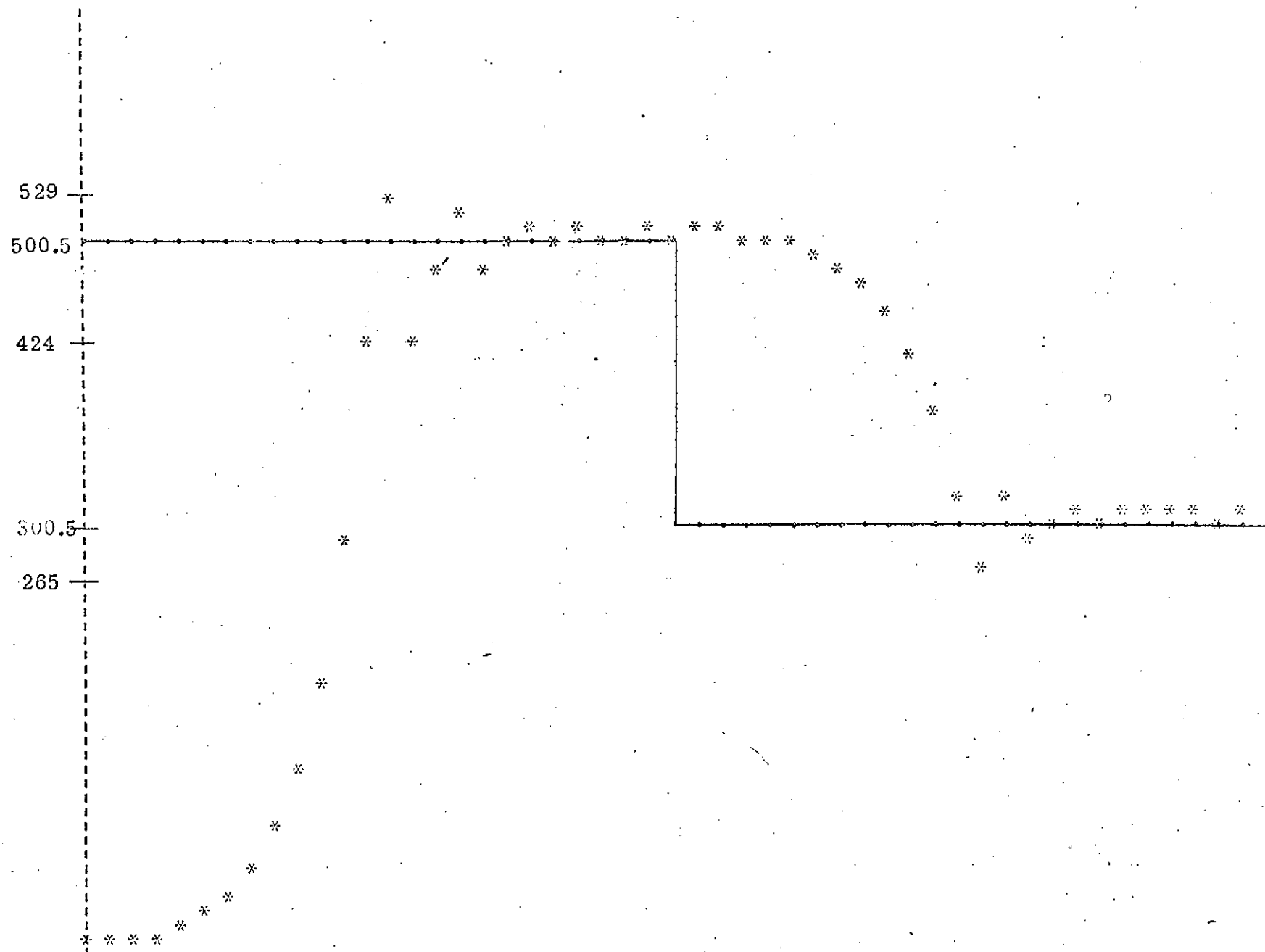


Fig. 7(b) The response of a Song-Adaptive Delta Modulator to abrupt level changes with Overshoot Suppression.

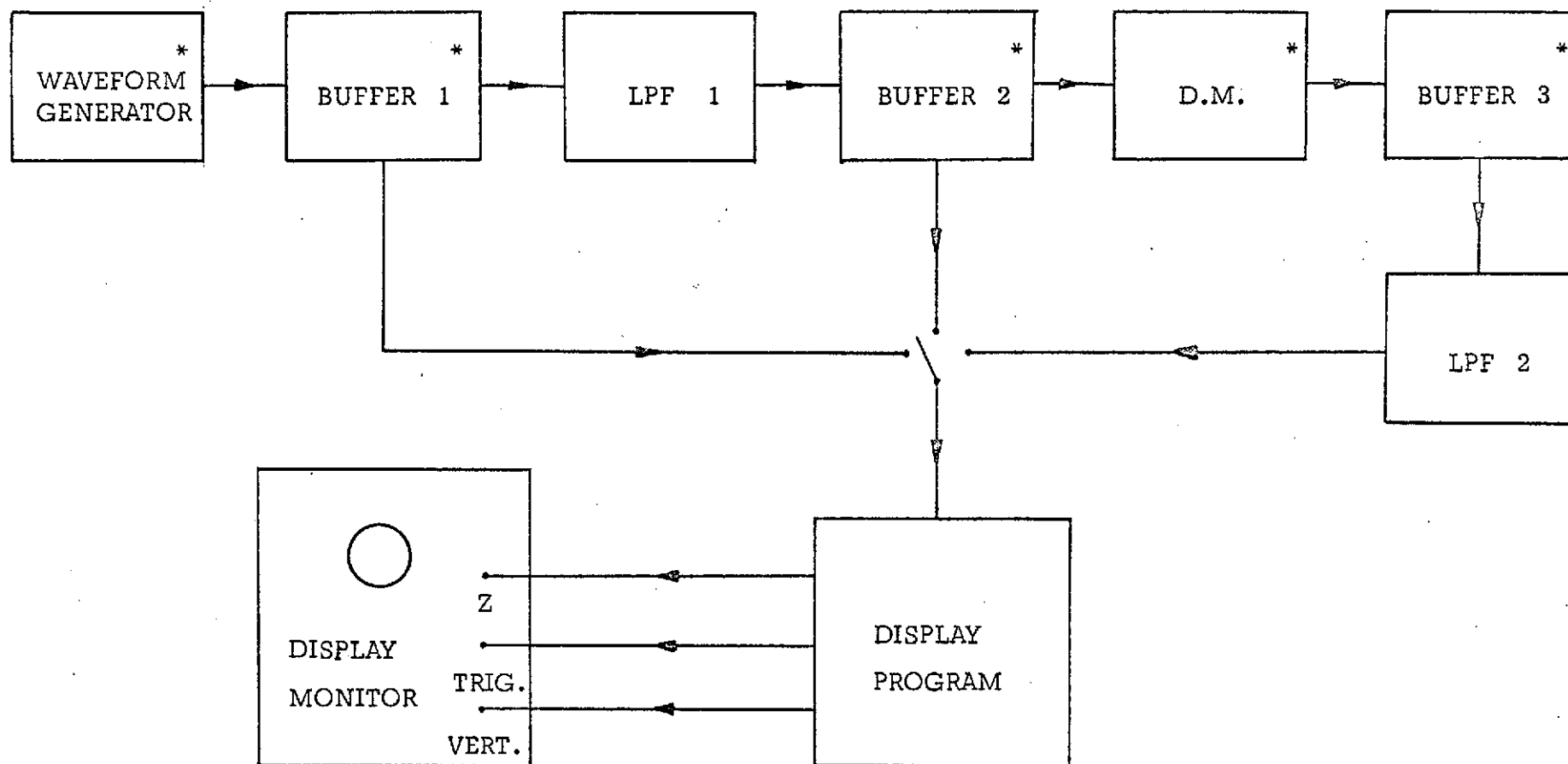


Figure 8 The Experimental Set Up.

Note: Blocks marked \* are stored and implemented in a PDP-8 computer.

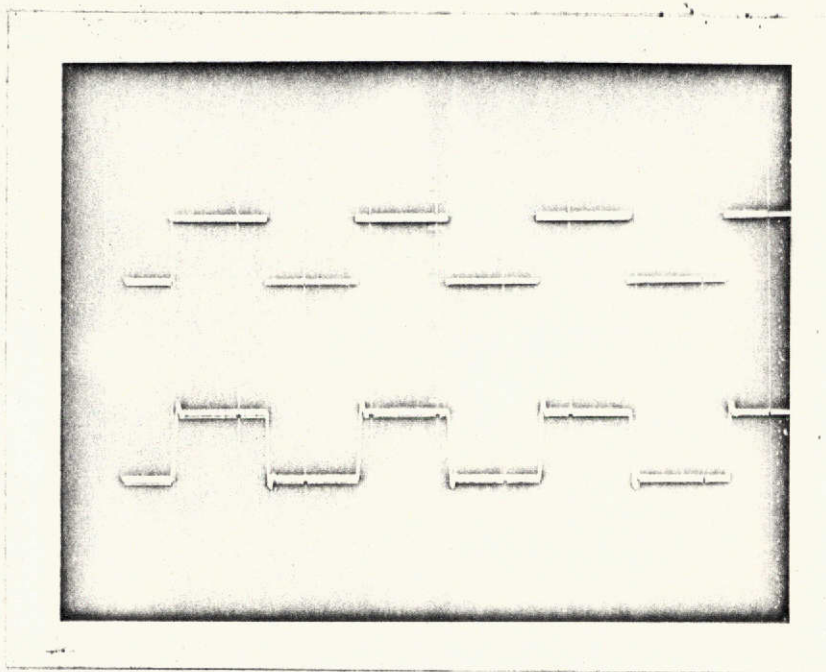


Figure 9(a) Upper Waveform: 350 Hz square wave ( $m(t)$ ).  
Lower Waveform:  $m(t)$  bandlimited to 10.5 KHz.

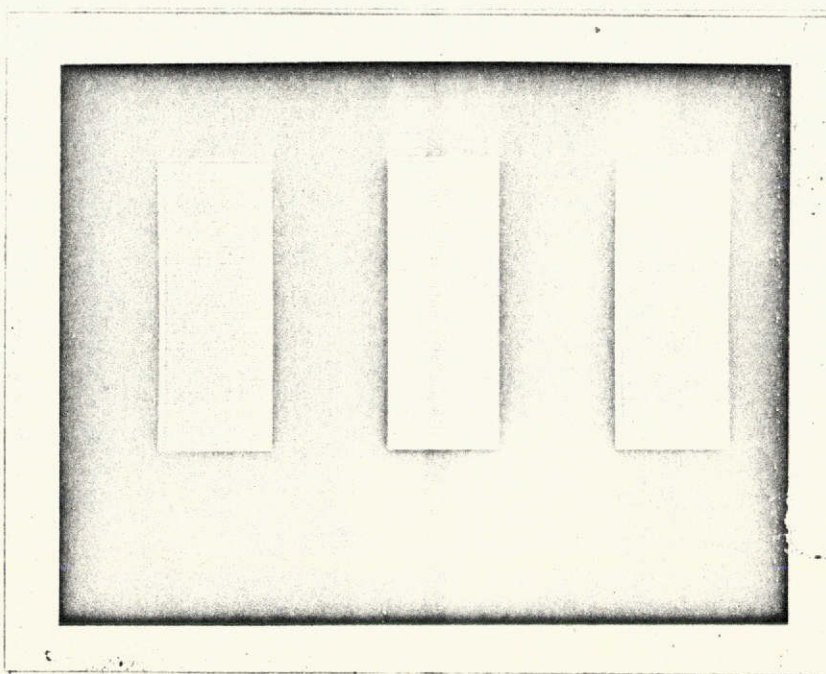


Figure 9(a') 100 line raster intensity modulated by bandlimited  $m(t)$ .

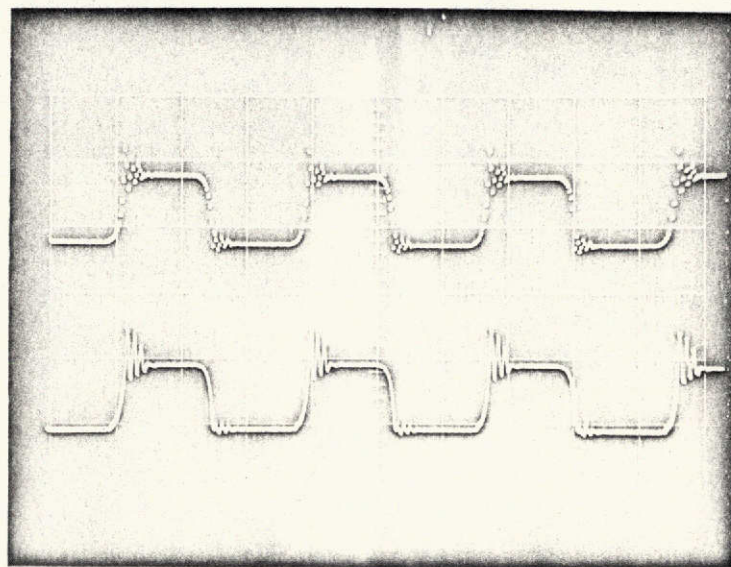


Figure 9(b) Upper Waveform: Bandlimited  $m(t)$  after DM processing (without O.S.S.) at  $f_s = 39.9$  KHz.  
 Lower Waveform: Upper Waveform low-pass filtered ( $f_c = 10.5$  KHz) =  $x(t)$

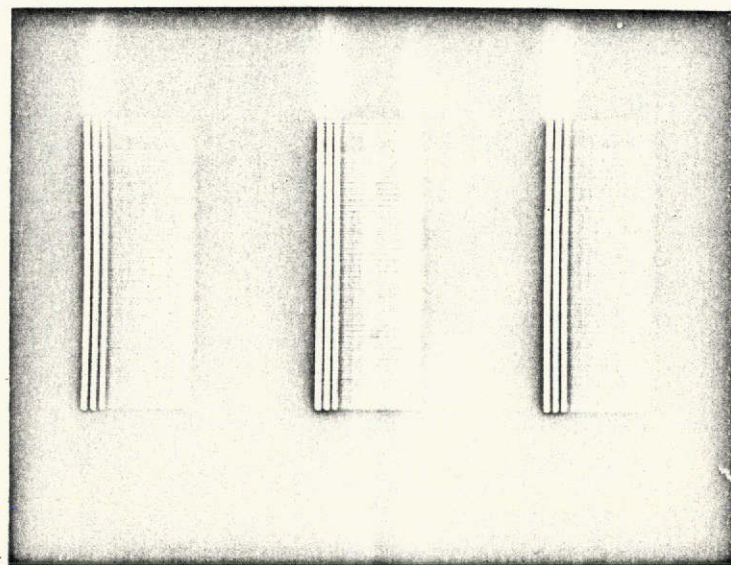


Figure 9b') 100 line raster intensity modulated by  $x(t)$ .

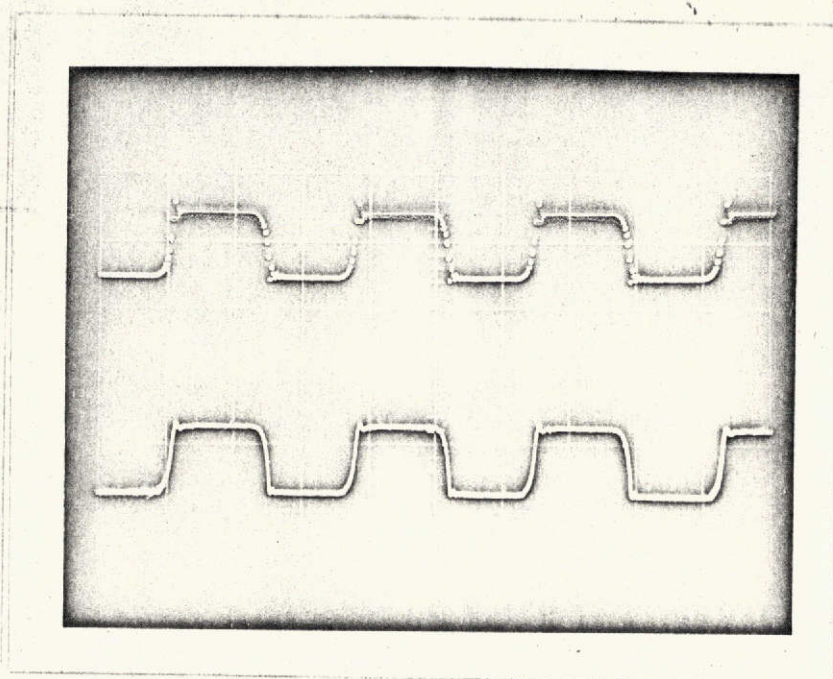


Figure 9(c) Upper Waveform: Bandlimited  $m(t)$  after DM processing (with O.S.S.) at  $f_s = 39.9$  KHz..

Lower Waveform: Upper Waveform low-pass filtered ( $f_c = 10.5$  KHz) =  $x'(t)$ .

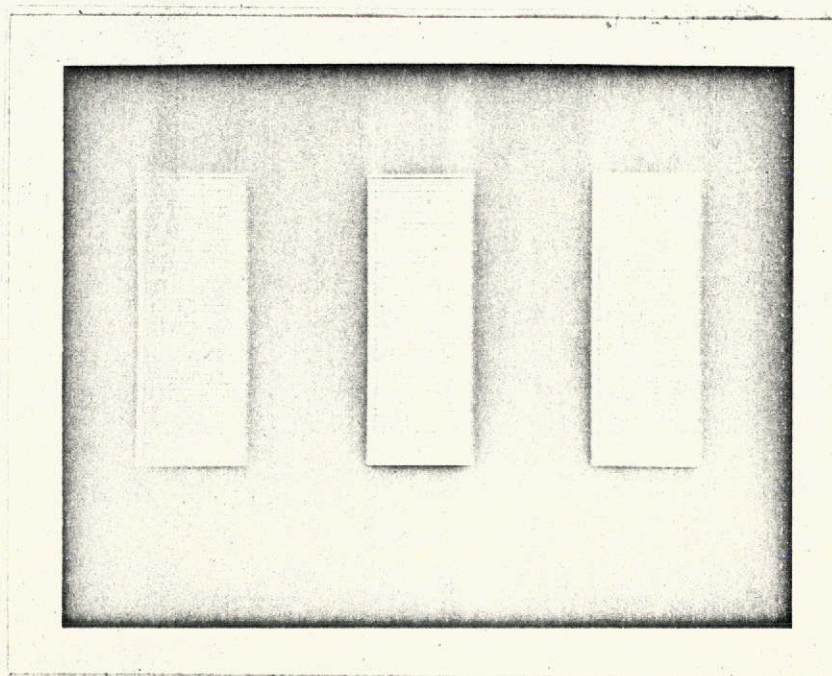


Figure 9(c') 100 line raster intensity modulated by  $x'(t)$ .

K= 0.000	SK= 500.5	XK= 0.000	EKE= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKE= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKE= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKE= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKE= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKE= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKE= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKE= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKE= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKE= 1.000
K= 10.00	SK= 500.5	XK= 189.0	EKE= 1.000
K= 11.00	SK= 500.5	XK= 283.0	EKE= -1.000
K= 12.00	SK= 500.5	XK= 424.0	EKE= 1.000
K= 13.00	SK= 500.5	XK= 635.0	EKE= -1.000
K= 14.00	SK= 500.5	XK= 936.0	EKE= -1.000
K= 15.00	SK= 500.5	XK= 1373.0	EKE= 1.000
K= 16.00	SK= 500.5	XK= 1951.0	EKE= 1.000
K= 17.00	SK= 500.5	XK= 2680.0	EKE= -1.000
K= 18.00	SK= 500.5	XK= 3510.0	EKE= -1.000
K= 19.00	SK= 500.5	XK= 4483.0	EKE= 1.000
K= 20.00	SK= 500.5	XK= 5660.0	EKE= 1.000
K= 21.00	SK= 500.5	XK= 7000.0	EKE= -1.000
K= 22.00	SK= 500.5	XK= 8530.0	EKE= 1.000
K= 23.00	SK= 500.5	XK= 10340.0	EKE= -1.000
K= 24.00	SK= 500.5	XK= 12460.0	EKE= -1.000
K= 25.00	SK= 500.5	XK= 14940.0	EKE= 1.000
K= 26.00	SK= 300.5	XK= 18000.0	EKE= -1.000
K= 27.00	SK= 300.5	XK= 21670.0	EKE= -1.000
K= 28.00	SK= 300.5	XK= 26030.0	EKE= -1.000
K= 29.00	SK= 300.5	XK= 31170.0	EKE= -1.000
K= 30.00	SK= 300.5	XK= 37100.0	EKE= -1.000
K= 31.00	SK= 300.5	XK= 43850.0	EKE= -1.000
K= 32.00	SK= 300.5	XK= 51460.0	EKE= -1.000
K= 33.00	SK= 300.5	XK= 60000.0	EKE= -1.000
K= 34.00	SK= 300.5	XK= 69500.0	EKE= -1.000
K= 35.00	SK= 300.5	XK= 80000.0	EKE= -1.000
K= 36.00	SK= 300.5	XK= 91500.0	EKE= 1.000
K= 37.00	SK= 300.5	XK= 104600.0	EKE= 1.000
K= 38.00	SK= 300.5	XK= 119500.0	EKE= -1.000
K= 39.00	SK= 300.5	XK= 136400.0	EKE= -1.000
K= 40.00	SK= 300.5	XK= 154900.0	EKE= 1.000
K= 41.00	SK= 300.5	XK= 175500.0	EKE= 1.000
K= 42.00	SK= 300.5	XK= 198400.0	EKE= -1.000
K= 43.00	SK= 300.5	XK= 223500.0	EKE= 1.000
K= 44.00	SK= 300.5	XK= 250800.0	EKE= -1.000
K= 45.00	SK= 300.5	XK= 280000.0	EKE= 1.000
K= 46.00	SK= 300.5	XK= 311200.0	EKE= -1.000
K= 47.00	SK= 300.5	XK= 344000.0	EKE= -1.000
K= 48.00	SK= 300.5	XK= 379000.0	EKE= 1.000
K= 49.00	SK= 300.5	XK= 416000.0	EKE= 1.000
K= 50.00	SK= 300.5	XK= 455000.0	EKE= -1.000

Table 1(a) Table of values for Fig. 7(a)

K= K'th sampling instant.

SK = input signal at time K.

XK = input estimate at time K.

EKE = sign bit at time K.

K= 0.000	SK= 500.5	XK= 0.000	EKU= 1.000	EKT= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKU= 1.000	EKT= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKU= 1.000	EKT= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKU= 1.000	EKT= 1.000
K= 4.000	SK= 500.5	XK= 15.000	EKU= 1.000	EKT= 1.000
K= 5.000	SK= 500.5	XK= 24.000	EKU= 1.000	EKT= 1.000
K= 6.000	SK= 500.5	XK= 37.000	EKU= 1.000	EKT= 1.000
K= 7.000	SK= 500.5	XK= 56.000	EKU= 1.000	EKT= 1.000
K= 8.000	SK= 500.5	XK= 84.000	EKU= 1.000	EKT= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKU= 1.000	EKT= 1.000
K= 10.000	SK= 500.5	XK= 189.0	EKU= 1.000	EKT= 1.000
K= 11.000	SK= 500.5	XK= 283.0	EKU= 1.000	EKT= 1.000
K= 12.000	SK= 500.5	XK= 424.0	EKU= 1.000	EKT= 1.000
K= 13.000	SK= 500.5	XK= 529.0	EKU= -1.000	EKT= -1.000
K= 14.000	SK= 500.5	XK= 424.0	EKU= 1.000	EKT= -1.000
K= 15.000	SK= 500.5	XK= 476.0	EKU= 1.000	EKT= 1.000
K= 16.000	SK= 500.5	XK= 515.0	EKU= -1.000	EKT= -1.000
K= 17.000	SK= 500.5	XK= 476.0	EKU= 1.000	EKT= -1.000
K= 18.000	SK= 500.5	XK= 425.0	EKU= 1.000	EKT= 1.000
K= 19.000	SK= 500.5	XK= 509.0	EKU= -1.000	EKT= -1.000
K= 20.000	SK= 500.5	XK= 495.0	EKU= 1.000	EKT= -1.000
K= 21.000	SK= 500.5	XK= 502.0	EKU= -1.000	EKT= -1.000
K= 22.000	SK= 500.5	XK= 499.0	EKU= 1.000	EKT= 1.000
K= 23.000	SK= 500.5	XK= 500.0	EKU= 1.000	EKT= 1.000
K= 24.000	SK= 500.5	XK= 501.0	EKU= -1.000	EKT= -1.000
K= 25.000	SK= 500.5	XK= 500.0	EKU= 1.000	EKT= -1.000
K= 26.000	SK= 300.5	XK= 502.0	EKU= -1.000	EKT= -1.000
K= 27.000	SK= 300.5	XK= 501.0	EKU= -1.000	EKT= -1.000
K= 28.000	SK= 300.5	XK= 499.0	EKU= -1.000	EKT= -1.000
K= 29.000	SK= 300.5	XK= 496.0	EKU= -1.000	EKT= -1.000
K= 30.000	SK= 300.5	XK= 492.0	EKU= -1.000	EKT= -1.000
K= 31.000	SK= 300.5	XK= 486.0	EKU= -1.000	EKT= -1.000
K= 32.000	SK= 300.5	XK= 477.0	EKU= -1.000	EKT= -1.000
K= 33.000	SK= 300.5	XK= 464.0	EKU= -1.000	EKT= -1.000
K= 34.000	SK= 300.5	XK= 445.0	EKU= -1.000	EKT= -1.000
K= 35.000	SK= 300.5	XK= 417.0	EKU= -1.000	EKT= -1.000
K= 36.000	SK= 300.5	XK= 375.0	EKU= -1.000	EKT= -1.000
K= 37.000	SK= 300.5	XK= 312.0	EKU= -1.000	EKT= -1.000
K= 38.000	SK= 300.5	XK= 265.0	EKU= 1.000	EKT= 1.000
K= 39.000	SK= 300.5	XK= 312.0	EKU= -1.000	EKT= 1.000
K= 40.000	SK= 300.5	XK= 289.0	EKU= 1.000	EKT= 1.000
K= 41.000	SK= 300.5	XK= 303.0	EKU= 1.000	EKT= 1.000
K= 42.000	SK= 300.5	XK= 308.0	EKU= -1.000	EKT= -1.000
K= 43.000	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= -1.000
K= 44.000	SK= 300.5	XK= 304.0	EKU= -1.000	EKT= -1.000
K= 45.000	SK= 300.5	XK= 302.0	EKU= -1.000	EKT= -1.000
K= 46.000	SK= 300.5	XK= 301.0	EKU= 1.000	EKT= 1.000
K= 47.000	SK= 300.5	XK= 302.0	EKU= -1.000	EKT= 1.000
K= 48.000	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= 1.000
K= 49.000	SK= 300.5	XK= 301.0	EKU= -1.000	EKT= -1.000

Table 1 (b) Table of values for Fig. 7(b)

EKU = sign bit used inside encoder and decoder.

EKT = sign bit transmitted from encoder to decoder.

## II. Rise Time Formulas and Truncation Errors

The ADM can be modeled as a nonlinear digital low-pass filter. The shortest rise times that can be handled by this model, without slope-overload, obviously depend on its parameters  $\Delta_o$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  (with  $\gamma = \alpha + \beta$ ). The speed with which the step-size increases is mainly a function of  $\gamma$ , higher values for  $\gamma$  will ensure shorter rise times and therefore are to be preferred in video transmission. Large values of  $\gamma$ , however, as already indicated, will lead to large overshoots and instabilities unless some O. S. S. scheme is used.

In what follows, in order to illustrate the complexity of the mathematical models needed for investigating this problem, and as a first step towards its solution, we derive a formula for the minimum rise time obtainable under given conditions. For this purpose, the rise time  $t_r$  (for brevity we will use in what follows the simplifying notation  $t_r = i$ ) is defined by the number of sampling instances needed in a given ADM (given  $\gamma$ ,  $\Delta_o$ ) in order to attain a specified voltage level  $V$ . Namely we look for a relation of the form

$$i = f_1(\gamma, V) ; \text{ with } \Delta_o = 1 \quad (1)$$

It is shown below that the main difficulty in deriving Eq (1) is due to the truncation errors inherent in the digital implementation, because these errors become too large to be neglected as  $i$  increases.

To derive Eq (1) we first derive an expression for

$$V = f_2(\gamma, i) \quad (2)$$

with  $\Delta_o = 1$  and assuming no truncation errors (infinite arithmetic capabilities)  
We obtain

$$V(i/\Delta_o=1, \text{no truncation error}) = 1 + 2 + 2\gamma + 2\gamma^2 + \dots + 2\gamma^{i-2} \quad (3)$$

which yields

$$V(i|\Delta_0=1, \text{no truncation error}) = 1 + 2 \left[ \frac{1 - \gamma^{i-1}}{1 - \gamma} \right] \quad ; \quad i \geq 3 \quad (4)$$

The values so obtained for  $V$  are upper bounds for its real values because the actual step sizes  $\Delta_K$  given by Eq (4), (Sec.I) are obtained with all fractional parts truncated. Let  $V_T(i)$  and  $V_{NT}(i)$  be the values of  $V(i)$  with and without considering the truncation error, respectively. Also, let  $\epsilon_{\text{cum}}(i)$  be the cumulative truncation error. We then have

$$V_T(i) = V_{NT}(i) - \epsilon_{\text{cum}}(i) \quad (5)$$

In computing  $\epsilon_{\text{cum}}(i)$  it is assumed that,  $r$ , the local round-off error is always the same and occurs only at every other sampling instant. Thus,

$$\epsilon_{\text{cum}}(i) = (r) + (\gamma r) + [\gamma(\gamma r) + r] + \langle r[\gamma(\gamma r) + r] \rangle + \dots \quad (6)$$

Therefore, the local truncation error  $\epsilon(K)$  is the solution of

$$\epsilon(k+1) = \gamma \epsilon(k) + \frac{r}{2} [1 + (-1)^k] \quad (7)$$

Equation (7) can be solved by taking the Z transform of both sides of the equation.

$$Z[\epsilon(k+1)] = \gamma Z[\epsilon(k)] + Z\left\{\frac{r}{2}[1 + (-1)^k]\right\} \quad (8)$$

Solving for  $Z(\epsilon(K))$

$$Z[\epsilon(k)] = \frac{rz^2}{(z-\gamma)(z-1)(z+1)} \quad (9)$$

Taking the inverse transform of (9) yields

$$\epsilon(k) = \frac{r\gamma^{k+1}}{\gamma^2-1} - \frac{r(-1)^k}{2(\gamma+1)} + \frac{r}{2(1-\gamma)} \quad (10)$$

Finally, the cumulative truncation error is given by

$$\begin{aligned} \epsilon_{cum}(i) = \sum_{k=1}^i \epsilon(k) &= \frac{r\gamma}{(\gamma^2-1)} \left( \frac{\gamma-\gamma^{i+1}}{1-\gamma} \right) - \frac{r[(-1)^i-1]}{4(\gamma+1)} \\ &\quad + \frac{ri}{2(1-\gamma)}, \quad i=1, 2, 3, \dots \end{aligned} \quad (11)$$

If the initial step  $\Delta(1) = 1$ , then it can be shown that  $\epsilon_{cum}(i) = 0$  for  $i < 4$ . Therefore, an index shift is introduced in (11), i.e., let  $i = n-3$ ,  $n \geq 4$ ,  $i \geq 1$ .

Thus,

$$\epsilon_{cum}(n) = \begin{cases} 0 & , n=1, 2, 3 \\ \frac{r\gamma}{(\gamma^2-1)} \left( \frac{\gamma-\gamma^{n-2}}{1-\gamma} \right) + \frac{r[1-(-1)^{n-3}]}{4(\gamma+1)} + \frac{r(n-3)}{2(1-\gamma)} & , n=4, 5, \dots \end{cases} \quad (12)$$

Using Eqs (4) and (12) in (5) we obtain,

$$V_T(i) \approx 1 + 2 \left[ \frac{1 - \gamma^{i-1}}{1 - \gamma} \right] - \frac{r\gamma}{(\gamma^2 - 1)} \left[ \frac{\gamma - \gamma^{i-2}}{1 - \gamma} \right] + \frac{r[(-1)^{i-3} - 1]}{4(\gamma + 1)} + \frac{r(3 - i)}{2(1 - \gamma)}, \quad i = 4, 5, \dots \quad (13)$$

Note that we may neglect the next to the last term in (13) because we always have  $r < 1$  and  $\gamma > 1$ . Next, (13) is solved for  $i$ , i.e., we find  $i = f(V, r, \gamma)$ . It is not possible however, to find an exact solution to (13) since it is a transcendental equation in  $i$ . Therefore, an iterative approach is used. Solving (13) for  $\gamma^i$  yields

$$\gamma^i = \frac{\gamma(\gamma^2 - 1)}{(r - 2\gamma^2 + 2)} \left[ (1 - \gamma)(V - 1) - \left( \frac{3r + 4}{2} \right) + \frac{r\gamma^2}{\gamma^2 - 1} + ri \right] \quad (14)$$

Therefore,

$$i = \frac{\ln \left\{ \frac{\gamma(\gamma^2 - 1)}{(r - 2\gamma^2 + 2)} \left[ (1 - \gamma)(V - 1) - \left( \frac{3r + 4}{2} \right) + \frac{r\gamma^2}{\gamma^2 - 1} + ri \right] \right\}}{\ln(\gamma)} \quad (15)$$

Equation (15) can be written as a recursion relation for  $i$ ,

$$i_{n+1} = \frac{\ln \left\{ \frac{\gamma(\gamma^2 - 1)}{(r - 2\gamma^2 + 2)} \left[ (1 - \gamma)(V - 1) - \left( \frac{3r + 4}{2} \right) + \frac{r\gamma^2}{\gamma^2 - 1} + ri_n \right] \right\}}{\ln(\gamma)} \quad (16)$$

If  $i_0 = 0$ , then:

$$i_1 = \frac{\ln \left\{ \frac{\gamma(\gamma^2-1)}{(r-2\gamma^2+2)} \left[ (1-\gamma)(\gamma-1) - \left( \frac{3r+4}{2} \right) + \frac{r\gamma^2}{\gamma^2-1} \right] \right\}}{\ln(\gamma)} \quad (17)$$

and

$$i_2 = \frac{\ln \left\{ \frac{\gamma(\gamma^2-1)}{(r-2\gamma^2+2)} \left[ (1-\gamma)(\gamma-1) - \left( \frac{3r+4}{2} \right) + \frac{r\gamma^2}{\gamma^2-1} + r i_1 \right] \right\}}{\ln(\gamma)} \quad (18)$$

If convergence occurs on the second iteration then Eq (18) combined with Eq (17) is the desired function,  $i = f(V, r, \gamma)$ . Computer calculations have shown that choosing  $r = 0.5$  ensures convergence by the second iteration and gives accurate results for all practical cases of  $V$  and  $\gamma$  that were tried. This is illustrated by Table 2 which shows that convergence occurs by the second iteration  $i_2$ . Furthermore, comparing XRONI, the exact level with truncation, ( $\gamma = 1.5$ ) at time  $I$  in Table 3 with the corresponding level (LEVEL) in Table 2, we see that the results for  $r = 0.5$  are accurate to the nearest sampling instant.

The rise time formula derived above is important for it specifies the rise time limitations of an ADM for a given  $\gamma$ . Considering the D. M. as a nonlinear low-pass filter, Eq (18) in effect specifies its bandwidth as a function of  $\gamma$ . Thus, in designing an ADM to handle a video signal band-limited to  $f_m$ , Eq (18) gives the minimum  $\gamma$  needed to prevent slope overload noise. Meanwhile, the O. S. S. algorithm described earlier is designed to prevent the overshoot and oscillations produced by the large  $\gamma$  that may be needed to handle the given signal.

THIS PROGRAM TABULATES LEVEL VS. RISETIME WITH GAMMA AND R  
AS PARAMETERS.

LEVEL=	1.00	I1=	1.36	I2=	0.86	I3=	0.51	I4=	0.57
LEVEL=	2.00	I1=	1.95	I2=	1.38	I3=	1.71	I4=	2.08
LEVEL=	3.00	I1=	2.42	I2=	1.84	I3=	2.11	I4=	2.02
LEVEL=	4.00	I1=	2.82	I2=	2.24	I3=	2.46	I4=	2.40
LEVEL=	5.00	I1=	3.17	I2=	2.60	I3=	2.78	I4=	2.74
LEVEL=	6.00	I1=	3.47	I2=	2.92	I3=	3.07	I4=	3.04
LEVEL=	7.00	I1=	3.74	I2=	3.21	I3=	3.33	I4=	3.31
LEVEL=	8.00	I1=	3.98	I2=	3.47	I3=	3.58	I4=	3.56
LEVEL=	9.00	I1=	4.20	I2=	3.71	I3=	3.80	I4=	3.79
LEVEL=	10.00	I1=	4.40	I2=	3.93	I3=	4.01	I4=	4.00
LEVEL=	11.00	I1=	4.59	I2=	4.14	I3=	4.21	I4=	4.20
LEVEL=	12.00	I1=	4.76	I2=	4.33	I3=	4.39	I4=	4.38
LEVEL=	13.00	I1=	4.92	I2=	4.50	I3=	4.56	I4=	4.55
LEVEL=	14.00	I1=	5.08	I2=	4.67	I3=	4.72	I4=	4.71
LEVEL=	15.00	I1=	5.22	I2=	4.83	I3=	4.87	I4=	4.87
LEVEL=	16.00	I1=	5.35	I2=	4.97	I3=	5.01	I4=	5.01
LEVEL=	17.00	I1=	5.48	I2=	5.11	I3=	5.15	I4=	5.15
LEVEL=	18.00	I1=	5.61	I2=	5.25	I3=	5.28	I4=	5.28
LEVEL=	19.00	I1=	5.72	I2=	5.37	I3=	5.40	I4=	5.40
LEVEL=	20.00	I1=	5.83	I2=	5.49	I3=	5.52	I4=	5.52
LEVEL=	21.00	I1=	5.94	I2=	5.61	I3=	5.64	I4=	5.63
LEVEL=	22.00	I1=	6.04	I2=	5.72	I3=	5.74	I4=	5.74
LEVEL=	23.00	I1=	6.14	I2=	5.83	I3=	5.85	I4=	5.85
LEVEL=	24.00	I1=	6.23	I2=	5.93	I3=	5.95	I4=	5.95
LEVEL=	25.00	I1=	6.32	I2=	6.03	I3=	6.04	I4=	6.04
LEVEL=	26.00	I1=	6.41	I2=	6.12	I3=	6.14	I4=	6.14
LEVEL=	27.00	I1=	6.50	I2=	6.21	I3=	6.23	I4=	6.23
LEVEL=	28.00	I1=	6.58	I2=	6.30	I3=	6.31	I4=	6.31
LEVEL=	29.00	I1=	6.66	I2=	6.38	I3=	6.40	I4=	6.40
LEVEL=	30.00	I1=	6.73	I2=	6.47	I3=	6.48	I4=	6.48
LEVEL=	31.00	I1=	6.81	I2=	6.54	I3=	6.56	I4=	6.56
LEVEL=	32.00	I1=	6.88	I2=	6.62	I3=	6.64	I4=	6.63
LEVEL=	33.00	I1=	6.95	I2=	6.70	I3=	6.71	I4=	6.71
LEVEL=	34.00	I1=	7.02	I2=	6.77	I3=	6.78	I4=	6.78
LEVEL=	35.00	I1=	7.08	I2=	6.84	I3=	6.85	I4=	6.85
LEVEL=	36.00	I1=	7.15	I2=	6.91	I3=	6.92	I4=	6.92
LEVEL=	37.00	I1=	7.21	I2=	6.98	I3=	6.99	I4=	6.99
LEVEL=	38.00	I1=	7.27	I2=	7.04	I3=	7.05	I4=	7.05
LEVEL=	39.00	I1=	7.33	I2=	7.11	I3=	7.12	I4=	7.12
LEVEL=	40.00	I1=	7.39	I2=	7.17	I3=	7.18	I4=	7.18
LEVEL=	41.00	I1=	7.45	I2=	7.23	I3=	7.24	I4=	7.24
LEVEL=	42.00	I1=	7.50	I2=	7.29	I3=	7.30	I4=	7.30
LEVEL=	43.00	I1=	7.56	I2=	7.35	I3=	7.35	I4=	7.35
LEVEL=	44.00	I1=	7.61	I2=	7.40	I3=	7.41	I4=	7.41
LEVEL=	45.00	I1=	7.66	I2=	7.46	I3=	7.47	I4=	7.47
LEVEL=	46.00	I1=	7.72	I2=	7.51	I3=	7.52	I4=	7.52
LEVEL=	47.00	I1=	7.77	I2=	7.57	I3=	7.57	I4=	7.57
LEVEL=	48.00	I1=	7.81	I2=	7.62	I3=	7.62	I4=	7.62
LEVEL=	49.00	I1=	7.86	I2=	7.67	I3=	7.67	I4=	7.67
LEVEL=	50.00	I1=	7.91	I2=	7.72	I3=	7.72	I4=	7.72

Table 2 LEVEL vs. RISETIME given by Eq.(16) for the first 4 iterations.

Note that for LEVEL > 5 convergence occurs by the second iteration, I2.

I= 1	XRONI= 1	XNRONI= 1.0000	ECUMI= 0.0000
I= 2	XRONI= 3	XNRONI= 3.0000	ECUMI= 0.0000
I= 3	XRONI= 6	XNRONI= 6.0000	ECUMI= 0.0000
I= 4	XRONI= 10	XNRONI= 10.5000	ECUMI= 0.5000
I= 5	XRONI= 16	XNRONI= 17.2500	ECUMI= 1.2500
I= 6	XRONI= 25	XNRONI= 27.3750	ECUMI= 2.3750
I= 7	XRONI= 38	XNRONI= 42.5625	ECUMI= 4.5625
I= 8	XRONI= 57	XNRONI= 65.3438	ECUMI= 8.3438
I= 9	XRONI= 85	XNRONI= 99.5156	ECUMI= 14.5156
I= 10	XRONI= 127	XNRONI= 150.7746	ECUMI= 23.7734
I= 11	XRONI= 190	XNRONI= 227.6600	ECUMI= 37.6602
I= 12	XRONI= 284	XNRONI= 342.9900	ECUMI= 58.9902
I= 13	XRONI= 425	XNRONI= 515.9850	ECUMI= 90.9854
I= 14	XRONI= 636	XNRONI= 775.4780	ECUMI= 139.4780
I= 15	XRONI= 952	XNRONI= 1164.7200	ECUMI= 212.7170
I= 16	XRONI= 1426	XNRONI= 1748.5800	ECUMI= 322.5760
I= 17	XRONI= 2137	XNRONI= 2624.3600	ECUMI= 487.3630
I= 18	XRONI= 3203	XNRONI= 3938.0500	ECUMI= 735.0440
I= 19	XRONI= 4802	XNRONI= 5908.5700	ECUMI= 1106.5700
I= 20	XRONI= 7200	XNRONI= 8864.3500	ECUMI= 1664.3500
*			

Table 3 Sampling Time vs. Level, with and without truncation, and Sampling Time vs. Cumulative Error ( $\gamma = \alpha + \beta = 1.5$ ).

Note: I = Sampling Instant

XRONI = Level at time I with truncation

XNRONI = Level at time I without truncation.

ECUMI = Cumulative error at time I

### III. Settling Time Calculations

Upper and lower bounds for the settling time of the oscillatory response of a Song DM are calculated in this section. Consideration is given to the DM with and without the overshoot suppression algorithm.

#### An Upper Bound - No O. S. S.

Referring to Fig 1(a) we see that the response of the Song DM to a voltage consists of rising steps until overshoot. After overshoot the voltage settles to the minimum step-size  $\Delta_o$ . The maximum settling time is calculated by assuming, as shown in the figure, that the decay occurs in steps of two; i.e.,  $x_k$  and  $x_{k+1}$  are each greater than  $m(t)$ ;  $x_{k+2}$  and  $x_{k+3}$  are each less than  $m(t)$ , etc.

It is clear that longer settling times would occur if the decay would occur in steps of three, four, etc.

The step-size prior to overshoot is  $\Delta_k$ , and at  $k+1, k+2, \dots$  is

$$\begin{aligned} |\Delta_{k+1}| &= (\alpha - \beta) \Delta_k \\ |\Delta_{k+2}| &= (\alpha + \beta) (\alpha - \beta) \Delta_k \\ |\Delta_{k+3}| &= (\alpha + \beta) (\alpha - \beta)^2 \Delta_k \end{aligned} \quad (1)$$

$$|\Delta_{k+N}| = \begin{cases} (\alpha + \beta)^{\frac{N-1}{2}} (\alpha - \beta)^{\frac{N+1}{2}} \Delta_k & , N \text{ odd} \\ (\alpha + \beta)^{\frac{N}{2}} (\alpha - \beta)^{\frac{N}{2}} \Delta_k & , N \text{ even} \end{cases} \quad (2)$$

$$(3)$$

We define the settling time  $N$  as the minimum time required for  $\Delta_{k+N} = \Delta_o$ .

Referring to Fig 1(a) we see that

$$\Delta_{k+2q} > \Delta_{k+2q+1} < \Delta_{k+2q+2} \quad (4)$$

Thus, we set  $\Delta_{k+N}$  in Eq 2 equal to the minimum step size  $\Delta_o$ . The minimum value of N (N odd) is then

$$N_i = 2 \frac{\ln \left[ \frac{\Delta_o}{\Delta_k} \sqrt{\frac{\alpha + \beta}{\alpha - \beta}} \right]}{\ln [\alpha^2 - \beta^2]} \quad (5)$$

#### An Upper Bound-Overshoot Suppression

Figure 1b) illustrates the response of the Song DM when overshoot suppression is employed. Here, the step at time k is

$$|\Delta'_k| = \frac{1}{2} \Delta_k \quad (6)$$

as a result of the suppression algorithm. Further,

$$|\Delta'_{k+1}| = \frac{1}{2} \Delta_k \quad (7a)$$

$$|\Delta'_{k+2}| = \frac{1}{2} (\alpha - \beta) \Delta_k \quad (7b)$$

$$|\Delta'_{k+3}| = \frac{1}{2} (\alpha + \beta)(\alpha - \beta) \Delta_k \quad (7c)$$

⋮

$$|\Delta'_{k+N}| = \frac{1}{2} (\alpha + \beta)^{\frac{N}{2}-1} (\alpha - \beta)^{\frac{N}{2}} \Delta_k, \quad N \text{ even} \quad (8)$$

Only the even value of N is given since

$$\Delta'_{k+2q-1} > \Delta'_{k+2q} < \Delta'_{k+2q+1} \quad (9)$$

as can be seen from Fig 1(b).

Setting  $\Delta_{k+N} = \Delta_o$  and solving for N yields

$$N_2 = 2 \frac{\ln \left[ 2(\alpha + \beta) \frac{\Delta_o}{\Delta_k} \right]}{\ln [\alpha^2 - \beta^2]} \quad (10)$$

#### Improvement When Using Overshoot Suppression

The improvement obtained when using overshoot suppression is

$$\Delta N = N_1 - N_2 = - \frac{\ln [4(\alpha^2 - \beta^2)]}{\ln [\alpha^2 - \beta^2]} \quad (11)$$

The minus sign occurs since  $\alpha^2 - \beta^2 < 1$ . Note that the improvement  $\Delta N$  is independent of  $\Delta_o$  and  $\Delta_k$ . If  $\alpha = 1$  and  $\beta = \frac{1}{2}$ ,  $\Delta N \approx 4$ .

#### A Lower Bound-No Overshoot Suppression

To obtain a lower bound we assume that the DM response continually oscillates about  $m(t)$  as shown in Fig 1(c). In this case

$$|\Delta_{k+1}| = (\alpha - \beta) \Delta_k \quad (12a)$$

$$\vdots$$

$$|\Delta_{k+N}| = (\alpha - \beta)^N \Delta_k \quad (12b)$$

Again setting  $|\Delta_{k+N}| = \Delta_o$ , yields

$$N_1 = \frac{\ln \left( \frac{\Delta_o}{\Delta_k} \right)}{\ln (\alpha - \beta)} \quad (13)$$

### A Lower Bound-Overshoot Suppression

The lower bound response with overshoot suppression is shown in Fig 1(d). In this case

$$|\Delta_{k+N}| = \frac{1}{2} (\alpha - \beta)^{N-1} \Delta_k \quad (14)$$

Setting  $\Delta_{k+N} = \Delta_o$ , yields

$$N_2 = \frac{\ln[2(\alpha - \beta) \frac{\Delta_o}{\Delta_k}]}{\ln[\alpha - \beta]} \quad (15)$$

If  $\frac{1}{2} \leq \alpha - \beta < 1$ ,  $N_2 \leq N_1$ . If  $\alpha = 1$ ,  $\beta = \frac{1}{2}$  there is no difference in the lower bound results.

In conclusion, we have bounded the settling time as

$$\frac{\ln(\frac{\Delta_o}{\Delta_k})}{\ln(\alpha - \beta)} \leq N \leq 2 \frac{\ln[\frac{\Delta_o}{\Delta_k} \sqrt{\frac{\alpha + \beta}{\alpha - \beta}}]}{\ln[\alpha^2 - \beta^2]} \quad \begin{array}{l} \text{no overshoot} \\ \text{suppression} \end{array} \quad (16)$$

$$\frac{\ln[2(\alpha - \beta) \frac{\Delta_o}{\Delta_k}]}{\ln(\alpha - \beta)} \leq N \leq 2 \frac{\ln[2(\alpha + \beta) \frac{\Delta_o}{\Delta_k}]}{\ln[\alpha^2 - \beta^2]} \quad \begin{array}{l} \text{overshoot} \\ \text{suppression} \end{array} \quad (17)$$

Thus, overshoot suppression reduces the settling time as well as the overshoot.

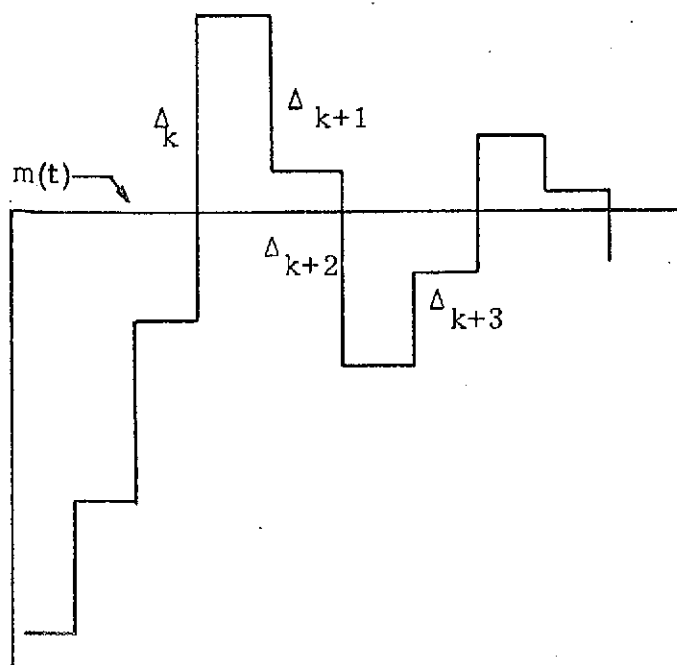


Figure 1(a) Step response without O.S.S. used to upper bound settling time.

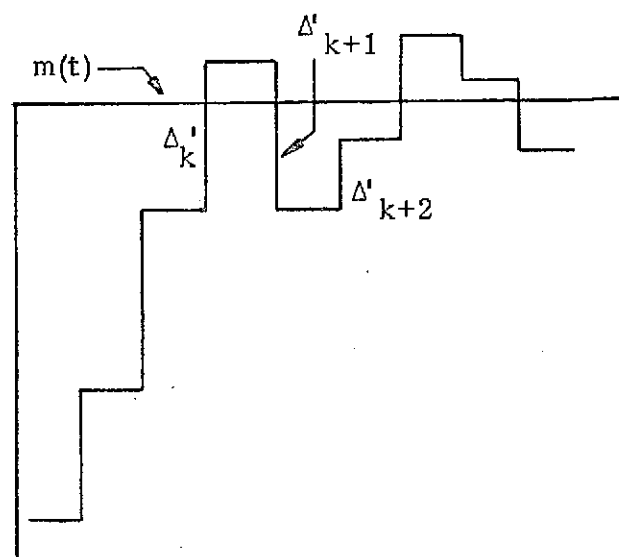


Figure 1(b) Step response with O.S.S. used to upper bound settling time.

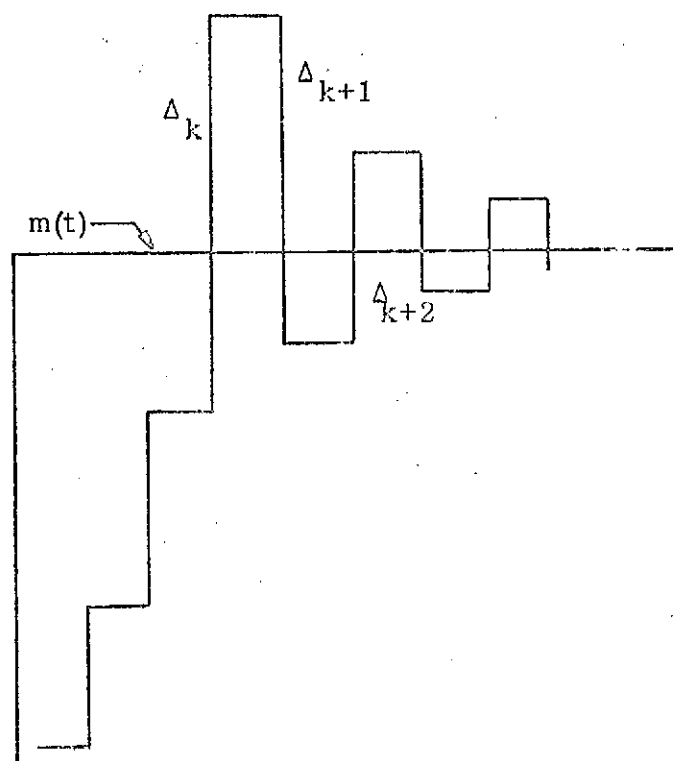


Figure 1(c) Step response without O.S.S. used to lower bound settling time.

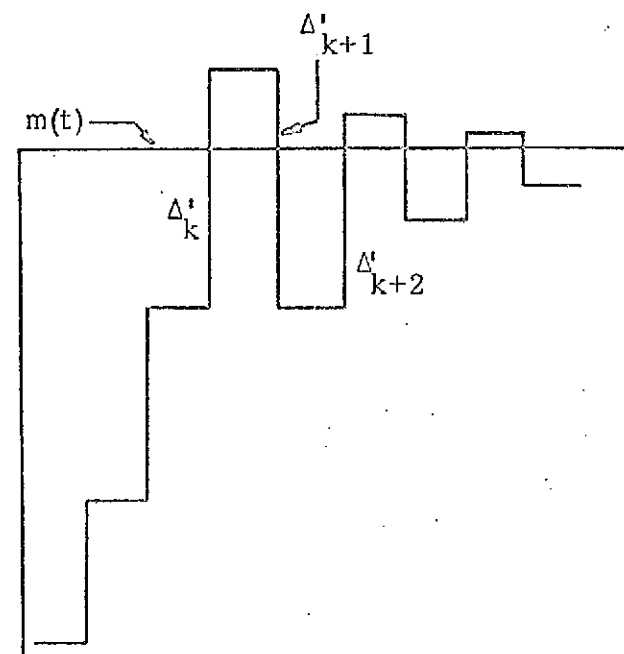


Figure 1(d) Step response with O.S.S. used to lower bound settling time.

#### IV. A Technique for Correcting Transmission Errors in Delta Modulation Channels

##### Introduction

The basic operation of a delta modulator may be explained briefly as follows. At the transmitter a signal is sampled periodically in time and compared to an estimated value, computed on the basis of previous history of the signal. The comparison yields a difference signal,  $d_k$ . The output,  $e_k$ , of the delta modulator will be a positive or a negative pulse depending on the sign of  $d_k$ . The output pulses,  $e_k$ , are transmitted over the communications channel.

If an error occurs in the polarity of  $e_k$ , during transmission, there will be a shift in the d.c. level of the received signal. These shifts in the d.c. level are cumulative in nature (1) and are undesirable in the video signal, because they produce bright or dark horizontal streaks throughout the picture. Different schemes to overcome this difficulty are proposed in the literature. They use spatial prediction (4), leaky integration (5), replacement of the erroneous lines by another line with an intensity which is the average over the intensities of the adjacent lines (6), transmission of a PCM word representing the absolute value of the signal sample after each horizontal line of the frame (3).

Because the effect of errors are more disturbing in nonlinear delta-modulated channels, we will describe our technique as implemented using Song's ADM (1). We briefly repeat below the main features of this ADM for completeness.

##### 1. The Song ADM

This ADM is shown in Fig 1 (inside the dashed lines). The analog signal,  $S(t)$ , is sampled at the  $k$ th instant of time, and converted into a digital signal,  $S_k$ . This signal is then compared to its estimate,  $X_k$ , obtained on the basis of previous samples. The output,  $e_k$ , is obtained from the relation

$$e_k = \text{sgn} (S_k - X_k) \quad (1)$$

and is transmitted over the channel. The estimate,  $X_k$ , is obtained using the recursive equation

$$X_k = X_{k-1} + \Delta_k \quad (2)$$

where  $\Delta_k$  is the new step size, given functionally by

$$\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1}) \quad (3)$$

Thus the new step size depends upon the previous step size,  $\Delta_{k-1}$ , and outputs,  $e_{k-1}$ , and  $e_{k-2}$ . The characteristics of the functions  $g_1$  and  $g_2$  are shown in Fig 2 indicating that

$$\Delta_k = \begin{cases} |\Delta_{k-1}|(\alpha e_{k-1} + \beta e_{k-2}) & , \quad |\Delta_{k-1}| \geq 2\Delta \\ 2\Delta e_{k-1} & , \quad |\Delta_{k-1}| < 2\Delta \end{cases} \quad (4)$$

where  $\alpha$ , and  $\beta$  are constants and,  $\Delta$ , is the minimum step size. The special region in Fig 2, where  $|\Delta_{k-1}| < 2\Delta$  is needed for preventing a "dead zone" at the origin. For video signals the values proposed by Song are  $\alpha = 1$ , and  $\beta = 0.5$ .

The decoder is just the feedback loop of the encoder and it reconstructs  $X_k$  from the  $e_k$  pulse train. The estimates are converted to an analog signal after passing them through a low pass filter to give  $\hat{S}(t)$ , the estimate of the transmitted signal at the receiver.

## 2. Square Wave Response:

A video signal has a large content of rapid level changes followed by regions of constant d.c. levels, and we therefore first present the response of the Song ADM to a square wave signal. Such a response is shown in Fig 3. The individual step sizes were calculated by using Equations (1) through (4), with  $\alpha = 1$  and  $\beta = \frac{1}{2}$ . With  $e_{k-1}$  and  $e_{k-2}$  of the same polarity, the new step

$\Delta_k$  will be 1.5 times the previous step size in the direction of  $e_{k-1}$ . When  $e_{k-1}$  and  $e_{k-2}$  are of opposite polarity the new step size will be one half the value of the previous step in the direction  $e_{k-1}$ . If the previous step size is less than twice the minimum step size, the new step size will be limited to  $2\Delta$  in the direction  $e_{k-1}$ . This will take the ADM out of the dead zone surrounding the origin.

A sharply rising signal will cause the ADM to produce a string of positive  $e_k$  pulses, while a falling signal will produce negative  $e_k$  pulses. For the region of constant d.c. level, a string of two positive pulses followed by two negative pulses will be produced until the level changes again.

### 3. Shifts in D.C. Levels:

The transmission noise inherent in every communication channel will introduce errors by reversing the polarity of  $e_k$ . Figure 3 exhibits, qualitatively how the d.c. level of the signal changes, when single or double errors occur. The solid line represents the signal at the decoder in the absence of the noise, while the broken line represents the signal corrupted by the noise. It is obvious from these figures that a shift in the d.c. level of the decoded signal occurs.

To estimate the shift in dc due to a single error, let the system be switched on at time  $k = 0$  with initial conditions;

$$\Delta_1 = 2\Delta \quad \text{and} \quad X_1 = 2\Delta$$

This yields

$$\Delta_2 = \left| 2\Delta \right| (\alpha e_1 + \beta e_0) \quad (5)$$

and

$$X_2 = \left| 2\Delta \right| (\alpha e_1 + \beta e_0) + 2\Delta. \quad (6)$$

By induction we can write the step size and estimate for the  $k$ th instant.

$$\Delta_K = \prod_{u=2}^{K-1} 2\Delta |\alpha e_{u-1} + \beta e_{u-2}| (\alpha e_{K-1} + \beta e_{K-2}) \quad (7)$$

and

$$X_K = \sum_{u=1}^K \Delta_u = \sum_{u=1}^{K-1} \prod_{u=1}^{u^*} 2\Delta |\alpha e_{u-1} + \beta e_{u-2}| (\alpha e_{K-1} + \beta e_{K-2}) \quad (8)$$

with the exception that for  $v = 1$ , the initial conditions hold.

Let  $e'_{k+1}$  be received instead of  $e_{k+1}$ , i.e.  $e'_{k+1} = -e_{k+1}$ . Then the erroneous estimate will be

$$X'_{k+1} = X_K + \prod_{u=1}^K 2\Delta |\alpha e_{u-1} + \beta e_{u-2}| (\alpha e'_{k+1} + \beta e_k) \quad (9a)$$

and

$$X'_{k+2} = X_K + \Delta'_{k+1} + \prod_{u=1}^{k+2} 2\Delta |\alpha e_{u-1} + \beta e_{u-2}| (\alpha e'_{k+1} + \beta e_k) \quad (9b)$$

The erroneous estimate at time interval  $N > k$  is

$$X'_N = X_K + \sum_{m=k+1}^N \Delta'_m \quad (10)$$

and the corresponding  $X_N$  using correct  $e_k$  will be given by

$$X_N = X_K + \sum_{m=k+1}^N \Delta_m \quad (11)$$

The d.c. shift at the  $N^{\text{th}}$  instant is

$$E_N = \sum_{m=k+1}^N (\Delta_m - \Delta'_m) \quad (12a)$$

and the relative d.c. shift  $\xi_N$  is given by:

$$\xi_N = \frac{E_N}{X_N} \quad (12b)$$

If all the  $e_k$ 's are in the same direction (i.e. either the signal is rising or falling) then it can be shown that

$$\xi_N = \frac{2\alpha^2(\alpha+\beta)^k \left[ 1 + 2\beta \sum_{n=0}^{N-2} (\alpha+\beta)^n \right] e'_{k+1}}{\sum_{v=1}^N \prod_{u=1}^v |\alpha e_{u-1} + \beta e_{u-2}| (\alpha e_v + \beta e_{v-1})} \quad (13)$$

In the steady state, there are two cases:

- a) when the error occurs after an alternating positive and negative pulse sequence, then;

$$\xi_{N1} = 2\Delta [(\alpha+\beta)^2(\alpha-\beta) \{(\alpha-\beta-1)(\alpha+\beta)-1\} + (\alpha+\beta)^2 + 2\beta + 2] e'_{k+1} \quad (14a)$$

- b) when the error occurs after two consecutive positive or negative pulses, then:

$$\xi_{N2} = 2\Delta [(\alpha+\beta-1)(\beta-\alpha) + (\alpha+\beta-1) + (1-\alpha^2+\beta^2) + 2\alpha] e'_{k+1} \quad (14b)$$

We also observe that the shift in the d.c. level due to erroneous  $e_k$  is cumulative in nature and therefore another error in the same direction will increase the shift. We also observe that two errors in opposite directions

may or may not have equal and opposite effects.

As an example of this, consider receiving a sequence of  $N$  positive pulses received by a delta modulator receiver having  $\alpha = 1$  and  $\beta = \frac{1}{2}$ . Assuming that this sequence followed a steady state sequence yielding 0 volts, we find the output voltages after  $N$  pulses to be

$$V_o(N) = \sum_{n=0}^{N-1} 2 \Delta (1.5)^n = 2 \Delta \frac{1 - (1.5)^N}{1 - (1.5)} \quad (15)$$

If, however, an error occurs on the  $k$ th pulse in the sequence, we can show that the output voltage  $V_{OE}(N)$  is now tightly bounded by

$$\sum_{n=0}^{K-1} 2 \Delta (1.5)^n + \sum_{n=0}^{N-K-2} 2 \Delta (1.5)^n \leq V_{OE}(N) \leq \sum_{n=0}^{K-1} 2 \Delta (1.5)^n + \sum_{n=0}^{N-K-1} 2 \Delta (1.5)^n \quad (16)$$

which can be written as

$$2 \Delta \left[ \frac{2 - (1.5)^K - (1.5)^{N-K+1}}{1 - (1.5)} \right] \leq V_{OE}(N) \leq 2 \Delta \left[ \frac{2 - (1.5)^K - (1.5)^{N-K}}{1 - (1.5)} \right] \quad (17)$$

If, we consider the worst-case condition, where  $\gamma \equiv V_{OE}(N)/V_o(N)$  is a minimum we find that the error should then occur at  $K = \frac{N+1}{2}$ , and the minimum ratio is then bounded by

$$\frac{2 - (1.5)^{N/2} [(1.5)^{1/2} + (1.5)^{-1/2}]}{1 - (1.5)^N} \leq \gamma_{\min} \leq \frac{2 - (1.5)^{N/2} 2 (1.5)^{1/2}}{1 - (1.5)^N} \quad (18)$$

If  $N \gg 1$ , this simplifies to

$$\gamma_{\min} \simeq (1.5)^{-N/2} \quad (19)$$

If, for example,  $N = 10$ , then  $K = 5$  for worst case conditions and

$$\{V_{DE}(N) - V_o(N)\}_{dB} = -18dB \quad (20)$$

This represents an extremely large shift in intensity level.

#### 4. The Error Correcting Algorithm

Adaptive Delta Modulation requires a lower bit rate than PCM to transmit the same information. However, channel errors produce errors which are cumulative in an adaptive DM system while independent in a PCM system. The basic idea involved in error correction is to send an occasional PCM word in lieu of several DM samples.

Consider then that a signal  $S(t)$  has a PCM sampling rate  $f_s$ . Then the DM will transmit  $N_D$  bits in a time  $T_s = 1/f_s$  so that the DM bit rate is  $f_s(D) = N_D f_s$ . After each  $i = kN_D$  bits are transmitted we insert a PCM word containing  $N_P$  bits. If these  $N_P$  bits are received without error then the received sequence is corrected. Using this algorithm the maximum length of error propagation is limited to  $i$  bits. We note that if  $N_P = N_D$  the system bit rate is not increased while if  $N_P > N_D$  the bit rate is increased by  $\frac{N_P - N_D}{i}$ . This is usually a small fraction and does not significantly increase the error rate.

This algorithm requires that the receiver be equipped with a PCM-DM converter. Such a converter is discussed in a Final Report for NASA-Goddard under Grant NGR 33-013-077.

An alternate algorithm, which does not require a PCM-DM converter merely compresses the  $i$  DM bits so that the  $N_P$  PCM bits can be inserted. This pulse stuffing algorithm increases the bit rate by  $\frac{N_P}{i}$  so that the bit rate

is now

$$f_s'(D) = \left(1 + \frac{N_p}{i}\right) f_s(D) \quad (21)$$

The value of "i" is chosen considering the probability of occurrence of the noise errors in the channel. Obviously there exists a trade-off between the increased error rate and the amount of improvement obtained in the quality of the received picture through the noisy channel. This trade-off can be used to meet the required specifications of bandwidth economy and picture quality.

Let the peak-to-peak voltage of "2V" volts be applied to the delta-modulator. If the word length used in the internal arithmetic of the delta-modulator is n-bits, the maximum number of possible quantization levels that can be simulate is  $2^n$ , and each level will represent " $\Delta$ " volts, where

$$\Delta = \frac{2V}{2^n} \quad (22)$$

A typical PCM word will use fewer bits than the number of bits used in the internal arithmetic of our system in order to minimize the bandwidth expansion needed. Let the PCM word be of length  $N_p = j$ . For such a system the gray level will be given by

$$1 \text{ gray level} = \frac{2V}{2^j} = Q \text{ volts} \quad (23a)$$

$$Q = \Delta \cdot 2^{n-j} \quad (23b)$$

##### 5. Generation of Errors in a Noiseless Channel

The substitution of this  $j < n$  bit PCM word will introduce an error in the absence of channel noise. This error can be calculated as follows: Assume for concreteness that 1's complement arithmetic is used. The voltages for  $n = 12$  are represented as follows

$$V^+ \equiv 011 \ 111 \ 111 \ 111$$

$$O^+ \equiv 000 \ 000 \ 000 \ 000$$

$$O^- \equiv 111 \ 111 \ 111 \ 111$$

$$V^- \equiv 100 \ 000 \ 000 \ 000$$

Note that there are two zeros, a positive zero and a negative zero. Consider that at a given time the estimate at the receiver is  $R_k = 111 \ 111 \ 111 \ 111$  ( $=O^-$ ) and the PCM word is  $000 \ 000 \ 000 \ 000$  ( $=O^+$ ), now consider that  $j = 0$ , i.e., the PCM word is represented by a single bit. Then the algorithm employed replaced the  $j$  most significant bits of  $R_k$  by the PCM word, which for our example yields

$$R_K^1 \equiv 011 \ 111 \ 111 \ 111 \equiv V^+$$

and  $R_K^1 - R_K = V^+$ . This is the maximum possible error. Note that this error could also be  $V^-$ . It can easily be verified that

- a) the number of such cases will be  $2^j$  (Let us call it a worst case.);
- b) the maximum level shift (error) due to substitution when there is no channel noise, in the above mentioned cases, will be given by,

$$\lambda_w = \pm \frac{2V}{2^j} \quad (24)$$

To illustrate this further, let  $j = 5$ ,  $i = 64$ , and  $n = 10$ . The bandwidth increase is then  $\frac{5}{71} = 7\%$ . The number of "worst case" possibilities is  $2^j = 32$ , and the relative d.c. shift due to each worst case will be  $\pm \frac{2V/2^j}{2V} = \pm \frac{1}{32} \times 100\% = \pm 3\%$ . In a system having 64 gray levels, this corresponds to a maximum error of 2 gray levels.

The d.c. shift is very small due to these errors, compared to the d.c. shift due to one ADM pulse in error which could be as much as  $\pm 18$  dB.

The error due to substitution in the absence of noise in the channel is even smaller than the d.c. shift due to an error in one ADM pulse occurring in the steady state. Also an error due to substitution is not commulative in nature, because it will correct itself during the next substitution period.

In conclusion our error correction algorithm differs in three respects with other solutions proposed in the literature.

- a) The correcting signal (not necessarily a full word) is inserted after,  $i, e_k$  pulses.
- b) The insertion rate is more frequent.
- c) Trade-off between increased bandwidth and the amount of correction obtained in the picture are made possible by changing the values of  $i$  and  $N_p = j$ .

## 6. Computer Simulation and Results

The delta-modulator including the correction algorithm and a random generator producing errors with programmable error rate was simulated on a PDP-8L digital computer. The results verify the expected d.c. level correction.

The correction algorithm is programmable for an arbitrary block length,  $i$ , and number of correcting bits,  $j$ . Results were obtained for,  $j = 2, 4, 6$  and full word substitution and for,  $i = 64, 128$ , and 256. The random error generator was programmed for a single error and two consecutive errors between correction. The two consecutive errors may occur in the same direction or the opposite direction depending upon the two  $e_k$ 's at the time of occurrence of the error.

The results of the computer simulation are shown in Fig 4(a) through (4f). The picture in Fig (4a) is the video reproduction through the ADM without noise. The picture in (4b) is the result of noise added to the channel. Note the cumulative nature of the d.c. shift. Figure (4c) shows

the effect of substitution with  $j = 4$  in the absence of noise in the channel while the picture in Fig (4d) shows the effect of substitution with  $j = 4$  and  $i = 64$  in the presence of noise. The picture in Fig (4e) was obtained with  $j = 6$  and  $i = 512$  while Fig (4f) represents a picture with full word substitution and  $i = 64$ .

## 7. Hardware Considerations

The correction circuitry based on the pulse stuffing technique is shown in Fig 1.

This hardware can be grouped into three different parts: a) the pulse-stuffing circuits (the Stuffer): b) the Counter: c) the rate changing circuits. This can be done in both the encoder and the decoder.

The pulse rate is increased in the encoder by a factor  $\Delta f = \frac{j}{i}$  and slowed down by the same amount in the decoder. The additional hardware needed in the implementation of both the encoder and the decoder, represents a small proportion of the total components that have to be added to those already included in the conventional adaptive delta modulator system. Also, both the modified and conventional DM include the same electronic subsystems, only used in slightly different ways.

Clearly this proposed technique will require an increase in the bandwidth required by the channel by the amount  $\Delta f$  mentioned above. This amount however, is of the order of 10% at most in our case ( $j = 2, 4, 6$ ;  $i = 32, 64, 128$ ). Moreover a trade-off exists between this increase in bandwidth and the improvement obtained in the error sensitivity of the channel. This trade off can be adjusted to meet any particular specifications, and requires only minor hardware modifications for every change.

## 8. Conclusions

The delta modulator is a relatively new device which has found wide application in voice encoding. Two main difficulties in using this device for video encoding and processing is that it slope overloads and "rings",

and that channel errors produce a d.c. level shift.

Recent work by Weiss (2) in the development of an easily implementable overshoot suppression algorithm has virtually eliminated the "ringing" problem.

In this report we have presented an algorithm, which is easily implemented, and which significantly reduces "picture streaks" due to the production of d.c. level shifts by channel errors. The algorithm can be made adaptive whereby the insertion rate of the PCM codeword needed for correction can be adjusted depending on the strength of the channel noise. This yields a more efficient use of bandwidth. Further, the resulting signal can still be processed further using line-to-line correlation to remove the remaining "streaks".

#### 9. References

- 1) C. L. Song; J. Garodnick; D. L. Schilling, "A Variable Step Size Robust Delta-Modulator" IEEE Transactions, Communications Technology, Vol. Com. - 19, pp 1033-1044, December 1971.
- 2) L. Weiss, I. Paz, D. L. Schilling, "Overshoot Suppression in Adaptive Delta-Modulator Links for Video Transmission", National Telecommunications Conference, November 26-28, 1973, pp 6D-1 to 6D-6.
- 3) E. J. Claire, "Bandwidth Reduction in Image Transmission" Radiation Incorporated Melbourne.
- 4) D. J. Connor, "Techniques for Reducing the Visibility of Transmission Errors in Digitally Encoded Video Signals" IEEE, Trans Comm. Technol, Vol. Com-21, June 1973.
- 5) H. S. McDonald and L. B. Jackson, "A Study of Single Frame Encoding of Pictures by Differential Pulse Code Modulation Incorporating a Lossy Integrator", Private Communications.
- 6) J. O. Limb, and F. W. Mounts, "Digital Differential Quantizer for Television", Bell Systems Tech. J. Vol. 48, pp 2583-2599, Sept. 1969.

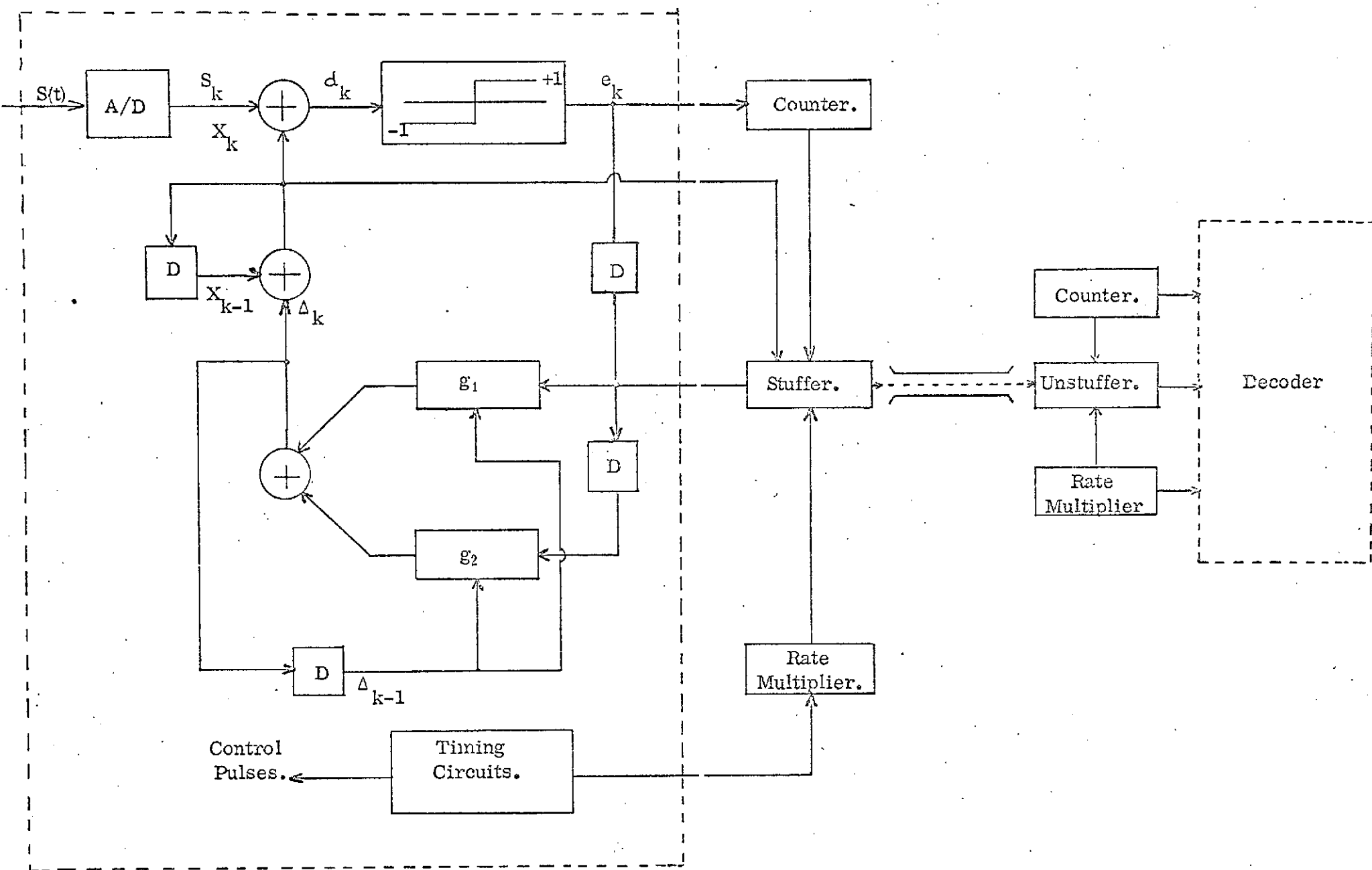


Fig. (1) Adaptive Delta Modulator with Correction Hardware. (Correction hardware is shown outside the dashed box).

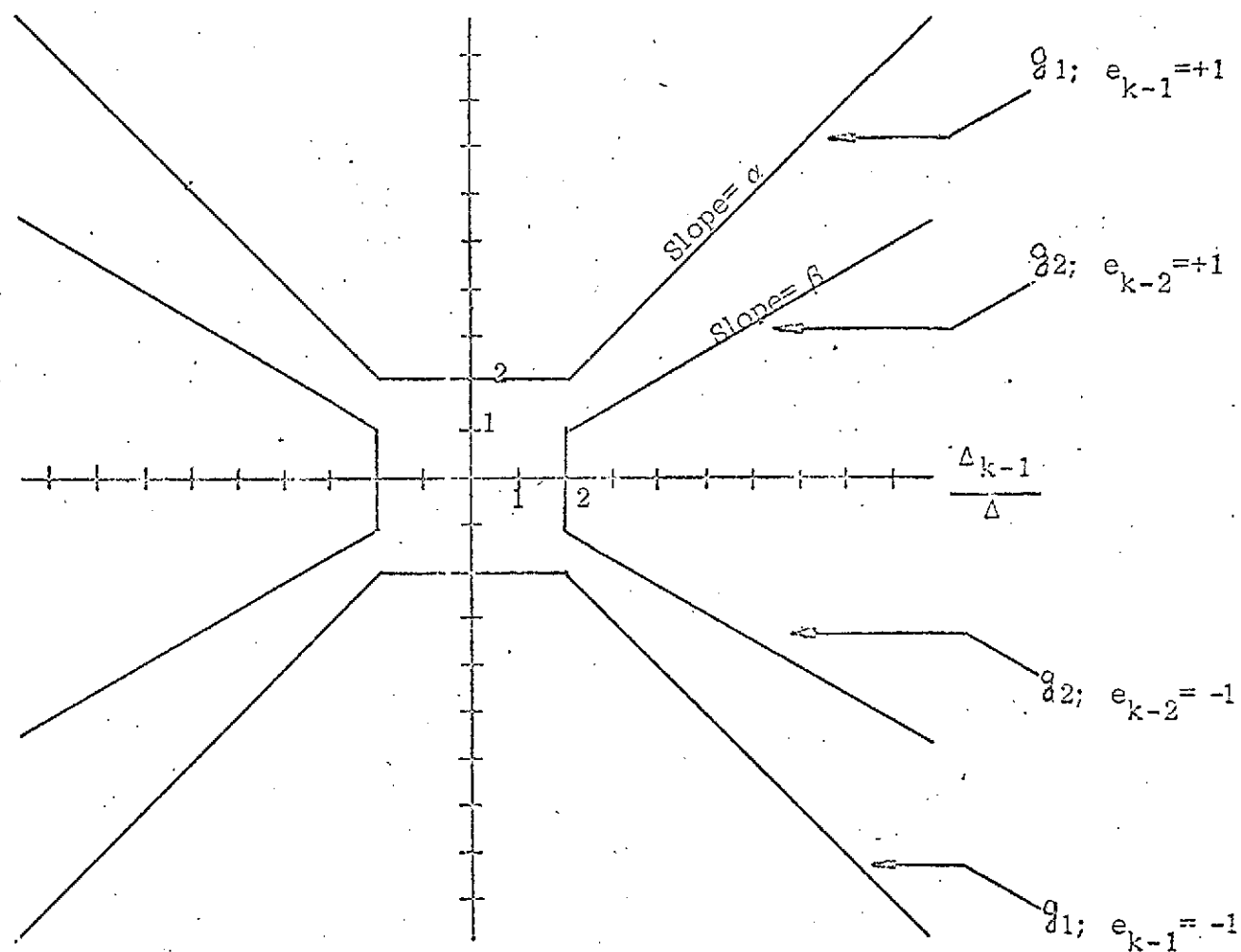


Fig. 2 Normalized  $q_1, q_2$  Characteristics. ( $\Delta$  = Minimum Step Size.)

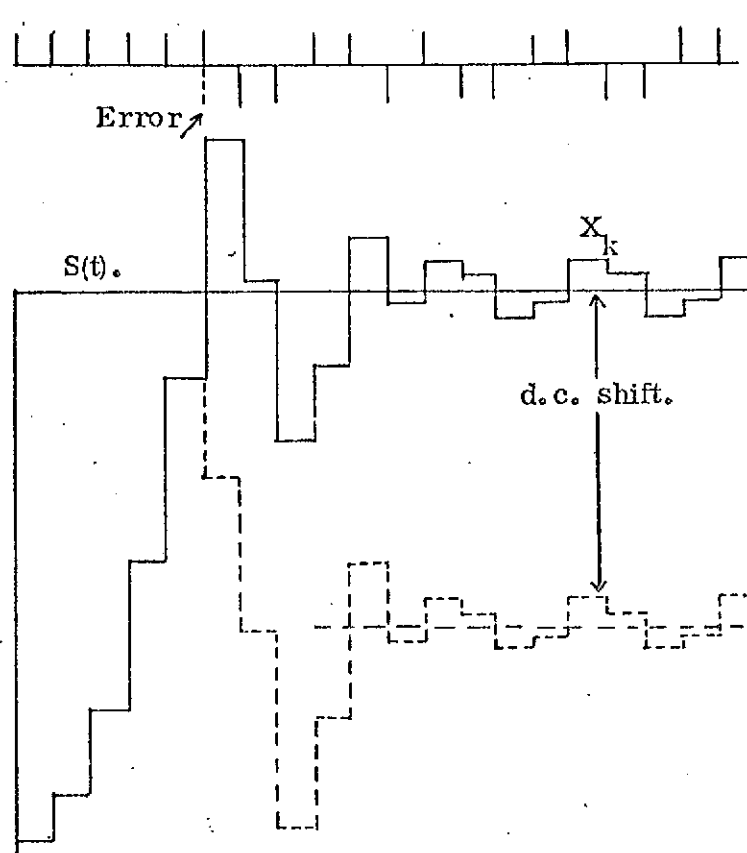


Fig. (3a) Effect of one error in  $e_k$  received at the decoder on rising  $e_k$  signal.

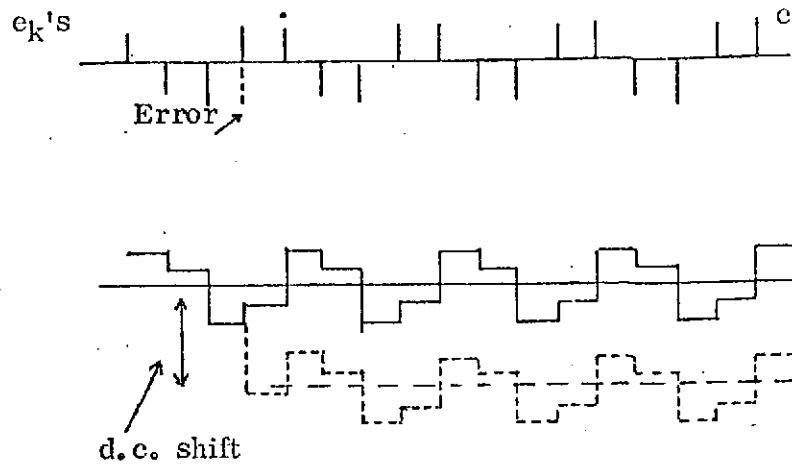


Fig. (3b) Effect of one error in the Steady state.

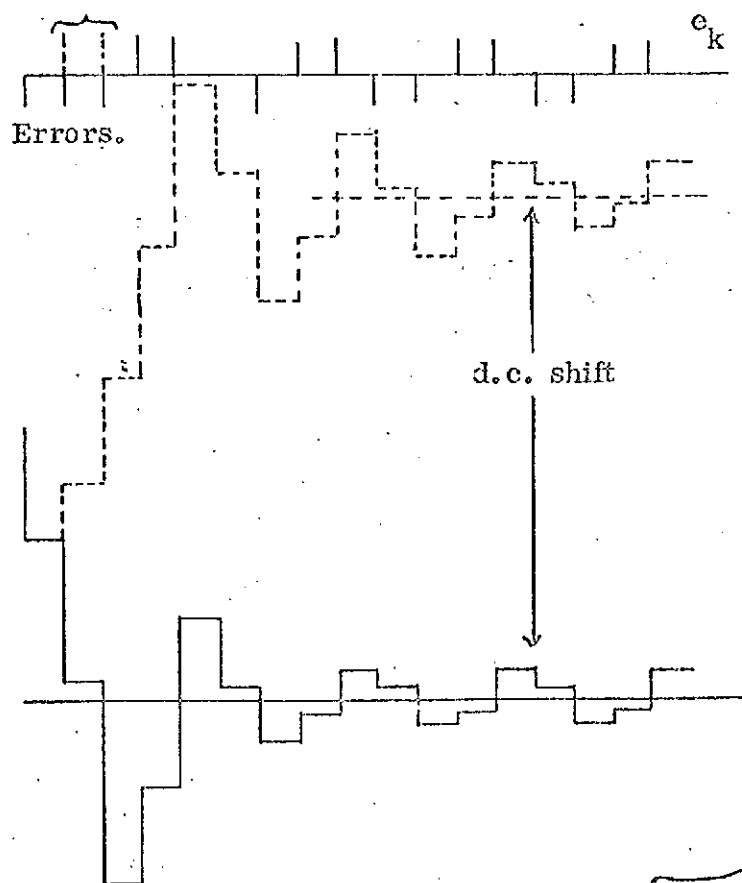


Fig. (3c) Effect of two errors on the falling signal.

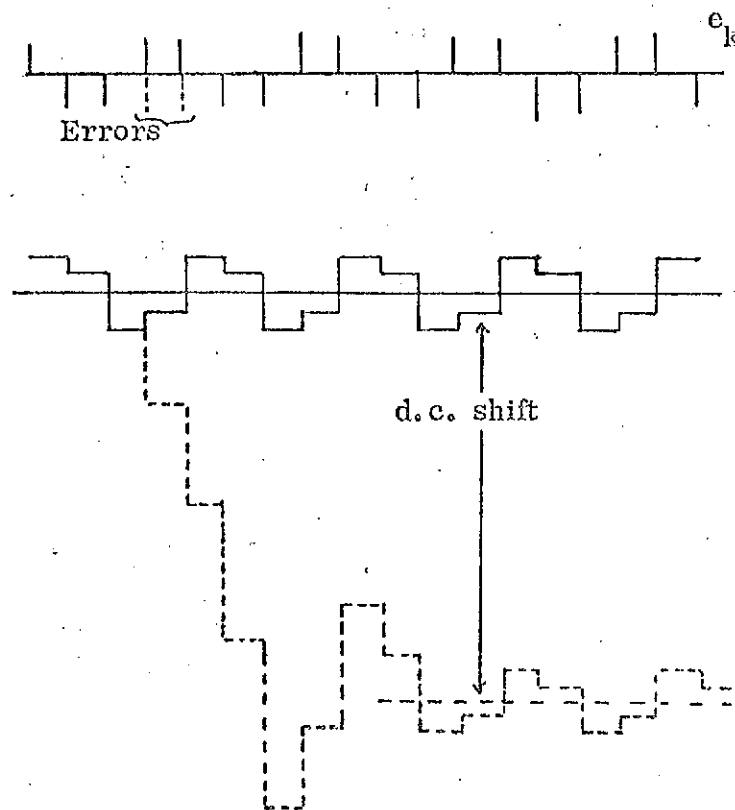


Fig. (3d) Effect of two errors in the steady state.

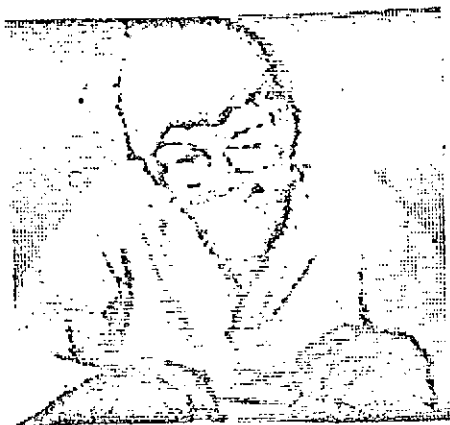


Fig.(4.a.) Picture recived through ADM, and Noiseless channel.

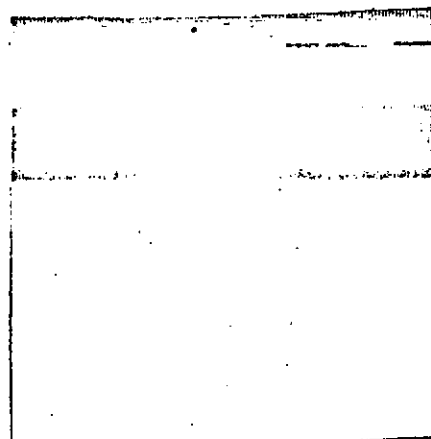


Fig.(4.b.) Picture decoded by ADM through noisy channel.

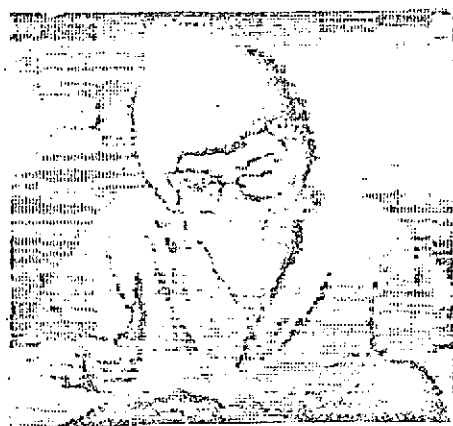


Fig.(4.c.) Picture through noiseless channel with 4-bit substitution.



Fig.(4.d.) Picture through noisy chann with 4-bit substitution.

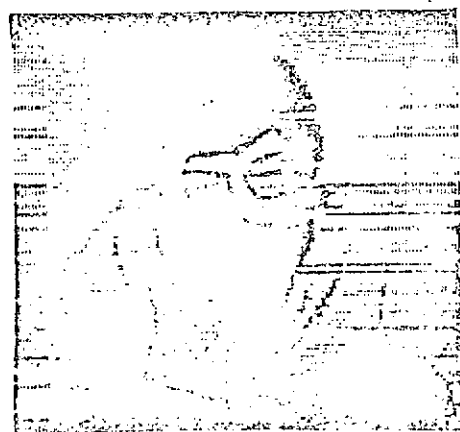


Fig.(4.e.) Picture through noisy channel.Six bit substitution after each line.

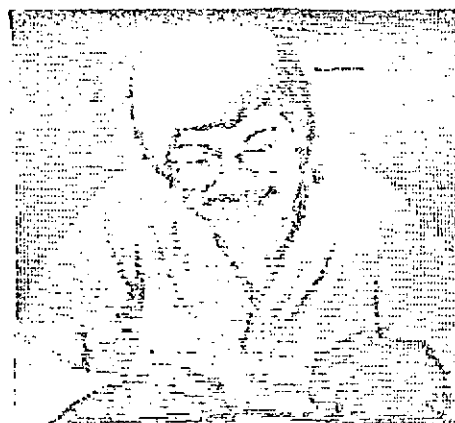


Fig.(4.f.) Full word substitution after a block length of 64 picture elements.

## V. Appendix

The material of this appendix has been taken from the Delta Modulation Status Report for the period of January 1, 1972 to July 1, 1972.

A stability condition is derived here assuming that  $\alpha > \beta$  and that in steady-state an oscillating pattern corresponding to the bit stream 11001100 . . . is produced. It is shown that for this pattern  $\alpha^2 - \beta^2 < 1$ . While other stable patterns are possible for different ranges of  $\alpha$  and  $\beta$  this pattern corresponds to the pattern of minimum settling time.

### 1. Stability of the Song Delta Modulator

The Song Delta Modulator shown in Fig 1 has been operated in two modes. These modes can best be described using Fig 2. Here we see that the output of function generator  $g_1$  is

$$g_1 = \begin{cases} |X_k - X_{k-1}| \operatorname{sgn}(e_k) & , \quad |X_k - X_{k-1}| \geq 2S \\ 2S \operatorname{sgn}(e_k) & , \quad |X_k - X_{k-1}| < 2S \end{cases} \quad (1)$$

where  $S$  is the minimum step-size.

The output of function generator  $g_2$  can be altered depending on whether voice or video is to be encoded. If voice is to be encoded we select

$$g_2 = \begin{cases} S \operatorname{sgn}(e_{k-1}) & , \quad |X_k - X_{k-1}| \geq 2S \\ 0 & , \quad |X_k - X_{k-1}| < 2S \end{cases} \quad (2)$$

If video is to be encoded

$$g_2 = \begin{cases} \frac{1}{2} |X_k - X_{k-1}| \operatorname{sgn}(e_{k-1}) & , \quad |X_k - X_{k-1}| \geq 2S \\ 0 & , \quad |X_k - X_{k-1}| < 2S \end{cases} \quad (3)$$

It is noted that for  $|x_k - x_{k-1}| \geq 2S$  the slope of  $g_1$  is unity while the slope of  $g_2$  is either zero or one-half. It is proven below that if the slope of  $g_1$  is  $\alpha$  and the slope of  $g_2$  is  $\beta$  then instability results if

$$\alpha^2 - \beta^2 \geq 1 \quad (4)$$

Proof: Referring to Fig 1 we see that

$$g_1(k) + g_2(k) = x_{k+1} - x_k \quad (5)$$

i.e., the values of  $g_1$  and  $g_2$ , when added, form the next value,  $x_{k+1} - x_k$ . For example, if  $x_1 - x_0 = 10S$  and  $e_k = +1$ , then  $g_1 = +10S$  and  $g_2 = +S$  (in the voice mode). Hence,  $g_1 + g_2 = 11S$  and  $x_2 - x_1 = 11S$ . Since  $x_k - x_{k-1}$  represents the instantaneous step-size, we see that in the voice mode the step-size increases or decreases by  $S$  providing  $|x_k - x_{k-1}| \geq 2S$ .

Proof that  $\alpha^2 - \beta^2 \geq 1$  Results in Instability

Assume that the slope of  $g_1$  is  $\alpha$  and the slope of  $g_2$  is  $\beta$ . Now refer to Fig 3. Here we see that the estimates  $x_{k-3}, x_{k-2}$ , etc., are approaching  $m(t)$ . If the system is stable,  $x_{k+1} < x_{k-3}$ . If the system is unstable  $x_{k-3} - x_{k+1} \leq 0$ . Since  $e_{k-3} = e_{k-2} = -1$ , we have from Fig 2,

$$|\Delta x_2| = (\alpha + \beta) |\Delta x_3| \quad (6)$$

Since  $e_{k-1} = +1$ ,

$$|\Delta x_1| = (\alpha - \beta) |\Delta x_2| \quad (7)$$

and similarly since  $e_k = +1$

$$|\Delta x_0| = (\alpha + \beta) |\Delta x_1| \quad (8)$$

Combining (6), (7) and (8) we have, for instability,

$$\begin{aligned}
 X_{K-3} - X_{K+1} &= |\Delta X_3| + |\Delta X_2| - |\Delta X_1| - |\Delta X_0| \\
 &= \left[ 1 + (\alpha + \beta) - (\alpha - \beta)(\alpha + \beta) - (\alpha - \beta)(\alpha + \beta)^2 \right] |\Delta X_3| \\
 &\leq 0
 \end{aligned} \tag{9}$$

Thus

$$(1 + \alpha + \beta)(1 - \alpha^2 + \beta^2) \leq 0 \tag{10}$$

Since  $\alpha, \beta \geq 0$  we have an unstable system if

$$1 - \alpha^2 + \beta^2 \leq 0 \tag{11a}$$

or

$$\alpha^2 - \beta^2 \geq 1 \tag{11b}$$

Hence, for stability

$$\alpha^2 - \beta^2 < 1 \tag{12}$$

Note that the procedure used for the Song mode:  $\alpha = 1$  and  $\beta = \frac{1}{2}$ , is stable.

The advantage of this system over other stable systems is that the step-size can increase by 50% ( $\alpha + \beta = 1.5$ ) and decrease by 50% ( $\alpha - \beta = 0.5$ ). This rapid rise and fall makes the Song mode of operation particularly suitable for video encoding.

## VI. New Areas of Research

### I. Calculation of Slope Overload Noise

Slope overload noise calculations have been made by Protonatarius for the linear DM. We are currently determining this slope overload noise for the ADM. In the ADM these calculations are complicated by the fact that the step size varies.

Another complication that we are considering is that the ADM operates in a discrete manner rather than analog. Protonatarius assumed that the estimate was continuous.

### II. New Overshoot Suppression Algorithm

In order to decrease the bit rate needed in a DM system we must have a system which can rise rapidly and settle quickly. The rate of rise of the Song DM is limited by  $\alpha + \beta$  where  $\alpha^2 - \beta^2 < 1$  is required for stability.

A new overshoot suppression algorithm is now being developed which would allow  $\alpha^2 - \beta^2 < 2$  and still yield a stable system.

## VII. Papers Published and Presented

1. "An All Digital Phase-Locked Loop for FM Demodulation," J. F. Greco and D. L. Schilling, pp. 43.37-43.41, ICC-73, June 1973.
2. "Overshoot Suppression in Adaptive Delta Modulator Links for Video Transmission", L. Weiss, I. Paz and D. L. Schilling, pp. 6D.1-6D.6, NTC-73, November 1973.
3. "Delta Modulation: With Applications to Video", D. L. Schilling, Seventh Hawaii Int'l Conference on System Sciences, January 1974.
4. "Technique for Correcting Transmission Errors in Delta Modulation Channels with Application to Video Signal Processing", Z. Ali, I. Paz, D. L. Schilling and N. Scheinberg, ICC-74, June 1974.
5. "Response of an All Digital Phase-Locked Loop,, J. Greco, J. Garodnick and D. L. Schilling, IEEE Trans. on Communications, June 1974.

# AN ALL DIGITAL PHASE-LOCKED LOOP FOR FM DEMODULATION

John Greco and Donald L. Schilling  
Department of Electrical Engineering  
City College of The City University of New York  
New York, New York 10031

## Abstract

An all digital phase-locked loop for FM demodulation is presented. The unit operates as a real time special purpose digital computer and employs a square wave voltage controlled oscillator. Design procedures are given for a first, second, and third order loop; the design reflects the influence of the square wave oscillator as well as quantization limitations. In an attempt to obtain information about the threshold of the digital phase-locked loop, the response to artificial input noise spikes is examined.

## 1. Introduction

The block diagram of the digital phase-locked loop (DPLL) is shown in Fig. 1. All signals to the right of the A/D converter are binary words and all computations within the DPLL are digital. The digital filter is either a proportional path, a proportional plus integral path, or a proportional plus integral plus double integral path yielding the first, second, or third DPLL respectively. To simplify the implementation, the filter gains are restricted to be  $1/2^N$ , with  $N$  an integer, so that coefficient multiplication is reduced to shifting the binary word. To avoid a binary multiplication of the input  $x_k$  and VCO output  $w_k$  at the phase detector, the digital VCO waveform is a square wave having values  $\pm 1$ . The multiplication is then reduced to a simple logic operation. An algorithm is developed (1) to determine the correct output of the digital VCO and an implicit VCO gain is introduced (1):  $G_{VCO} = \pi 2^N/5$ ,  $N$  = integer.

## 2. Digital Phase-Locked Loop Equation

To avoid distortion in the sampled incoming FM signal, the sampling frequency,  $f_s$ , is chosen according to

$$f_s = 2m B \quad (1a)$$

$$nf_s = f_0 - B/2 \quad (1b)$$

where  $B$  = IF bandwidth

$$m \geq 1$$

$$n = \text{integer}$$

$$f_0 = \text{carrier frequency.}$$

The VCO output is given by

$$w_k = \text{Sq} \left( k \frac{\pi}{2m} + \hat{\phi}_k \right) \quad (2)$$

where  $\hat{\phi}_k$  = VCO phase,

$$\text{Sq}(x) = \begin{cases} +1, & 0 \leq x < \pi \\ -1, & \pi \leq x < 2\pi \end{cases}$$

and  $\text{Sq}(x)$  is periodic with period  $2\pi$ . The DPLL output  $y_k$  is the derivative of the VCO phase  $\hat{\phi}_k$ ; in our digital scheme the derivative is computed as the difference:

$$y_k = \hat{\phi}_{k+1} - \hat{\phi}_k \quad (3)$$

Finally, the output  $y_k$  is determined from the phase error signal  $e_k$  by the digital filter characteristic. For the first order DPLL,  $y_k = g e_k$  and combining Eqs (2) and (3), we obtain the difference equation for the VCO phase:

$$\hat{\phi}_{k+1} = \hat{\phi}_k - 2G \cos \left( k \frac{\pi}{2m} + \phi_k \right) \text{Sq} \left( k \frac{\pi}{2m} + \hat{\phi}_k \right) \quad (4)$$

where  $G$  = loop gain. This nonlinear difference equation is extremely difficult to analyze and so a linearized model is introduced.

## 3. Linearized Model

The function  $\text{Sq}(\cdot)$  contains odd harmonics and the product of input and square wave produces harmonics at  $0, f_s/2m, 2f_s/2m, \dots, Nf_s/2m, \dots$  Hz. Of course, the zero harmonic is the only useful term as it contains the phase error information. Hence, we would like to have the other harmonics fall outside the DPLL bandwidth. But this is impossible as the  $(2m)^{\text{th}}$  harmonic is at  $f_s$  Hz which is equivalent to 0 Hz. That is, the aliasing produced by sampling introduces a term at 0 Hz. This term has amplitude on the order of  $1/4m$ , and so it appears that we should choose  $m$  large. However, choosing  $m$  large results in the harmonic at  $f_s/2m$  being introduced into the DPLL bandwidth. The constraint we impose is to have this harmonic fall outside the DPLL bandwidth. If the input frequency deviation is  $\Delta f$ , the bandwidth of this harmonic is approximately  $4 \Delta f$ , and so the constraint is

$$f_s / 2m - 2 \Delta f \geq B_L \quad (5)$$

with  $B_L$  = DPLL bandwidth.

If we can choose  $m \geq 3$ , the  $(2m)^{\text{th}}$  harmonic will be at least 20 dB below the fundamental and its contribution to the output and VCO phase can be neglected.

With Eq (5) satisfied we introduce the linearized model of the DPLL by using the VCO fundamental to generate the phase error. This fundamental has amplitude  $4/\pi$ , and assuming  $|e_k| < \pi/2$ , we arrive at the linearized model shown in Fig. 2, where  $G_{VCO}$  = implicit VCO gain.

## 4. Phase Error Constraint

The phase error  $e_k$  should be small so that we operate on the linear portion of the phase characteristic; i.e.,  $\sin e_k \approx e_k$ . The error in this approximation is referred to as harmonic distortion. However,  $e_k$  is represented by a finite number of bits and a small  $e_k$  will be lost in the quantization noise. Hence, an optimum phase error exists for which signal-to-harmonic distortion and signal-to-quantization noise are equal. These quantities are computed (assuming  $e_k$  uniformly distributed over its range) and plotted against  $(e_k)_{\text{max}}$  for 10-bit quantization and 12-bit quantization in Fig. 3. The intersections give the optimum value of  $(e_k)_{\text{max}}$ :

Number of bits	$(e_k)_{\text{max}}$ (volts)
10	0.35
12	0.23

## 5. Design of the First Order DPLL

An input frequency offset of  $\Delta f$  produces a phase error

$$e_k = \frac{2\pi \Delta f / f_s}{G} \quad (6)$$

The DPLL is constructed using 10-bit arithmetic; there-

fore

$$\frac{2\pi \Delta f / f_s}{G} < 0.35 \quad (7)$$

The DPLL bandwidth  $B_L$  is given approximately by

$$B_L \approx f_s \frac{2}{\pi^2} G \quad (8)$$

where the approximation is valid for  $\frac{4}{\pi} G \ll 1$ . Utilizing Eq. (8) in Eq. (5),

$$\frac{f_s}{2m} - 2\Delta f \geq f_s \frac{2}{\pi^2} G \quad (9)$$

Combining with Eq. (7) yields the design equation

$$m \leq \frac{f_s}{11.28 \Delta f} \quad (10)$$

The DTL and TTL logic employed necessitated the value  $f_s = 50$  kHz to provide a sufficient computation interval. Choosing  $\Delta f = 600$  Hz we obtain  $m \leq 7.4$ . We shall use  $m = 8$  (the VCO algorithm is simplified when  $m$  is a power of 2). We further obtain from Eq. (7)  $G \geq 0.214$  and  $B_L = 2.14$  kHz. The choice  $m = 8$  invalidates Eq. (5); we shall choose

$$G = 0.157 = \pi / 5 \cdot 2^3 \quad (11)$$

and allow  $(e_k)_{\max} = 0.48$ . Then  $B_L = 1.57$  kHz (the approximation of Eq. (8) is conservative; the actual bandwidth is 1.8 kHz) and Eq. (5) is satisfied.

Summarizing, the parameters for the first order DPLL are:

$$\begin{aligned} f_s &= 50 \text{ kHz} \\ \Delta f &= 600 \text{ Hz} \\ G &= 0.157 = \pi / 5 \cdot 2^3 \end{aligned}$$

#### 6. Design the Second Order DPLL

The digital filter is augmented to a proportional plus integral filter having transfer function  $H(z)$  given by

$$H(z) = g_1 + g_2 \frac{1}{1 - z^{-1}} \quad (12)$$

where  $g_1$  = proportional path gain

$g_2$  = integral path gain.

We shall use the same sampling frequency and frequency deviation computed for the first order DPLL.

The second DPLL tracks a frequency offset with zero phase error. Therefore we consider sinusoidal modulation:  $\hat{\phi}_k = \beta \sin k 2\pi f_m / f_s$ . If  $f_m \ll f_s$ , the phase error amplitude is

$$(e_k)_{\max} \approx \frac{2\pi \Delta f / f_s}{G_2} 2\pi f_m / f_s \quad (13)$$

where  $G_2$  = gain of the integral loop. Choosing  $f_m = 200$  Hz ( $\beta = 3$ ) we obtain from the phase error restriction  $G_2 \geq 0.0055$ . We choose initially  $G_2 = 0.00818 = \pi / 5 \cdot 2^6$ .

Now we have  $m$  and  $G_1$  at our disposal to meet Eq. (3). But we wish to achieve more than simply satisfying this equation—we seek a second order DPLL which provides threshold improvement over the first order DPLL. To gain insight into the loop's behavior near threshold, we introduce an artificial spike to the DPLL and observe the response (VCO phase). Obviously, if the VCO phase follows the input spike, the spike is reproduced at the output.

The model used for the spike is a sinusoidal increase of  $2\pi$  radians in the input phase:

$$(\phi_k)_{\text{slope}} = 2\pi \left( \frac{1}{2} - \frac{1}{2} \cos k\pi / K \right) \quad (14)$$

The spike duration,  $K/f_s$ , is chosen as the reciprocal of the IF bandwidth; this is the fastest spike possible. This spike is superimposed on constant or sinusoidal modulation, producing a positive spike when the input frequency is at the left extreme of the IF bandwidth. The nonlinear difference equation is solved on a computer for the VCO phase and the solution is examined to determine whether or not the loop follows the spike. Note that the linearized model is not valid when a spike appears as the phase error becomes large. Also note that the carrier amplitude remains constant during the spike; this represents a worst case as in actuality the amplitude decreases.

The results are displayed in Fig. 4, where the spike responses are displayed as a function of loop gains. Fig. 4 (a) is the case of the spike superimposed upon an input frequency deviation of 600 Hz; Fig. 4 (b) is for sinusoidal modulation with  $f_m = 200$  Hz and  $\beta = 3$ . They clearly show that  $m = 8$  is unsatisfactory as the loop becomes unstable for almost all values of  $G_1$ . The case  $m = 4$  is not much better. If we decrease the integral gain to  $\pi / 5 \cdot 2^4$  (and consequently increase the phase error) then we can choose  $G_1 = \pi / 5 \cdot 2^4$  or  $\pi / 5 \cdot 2^5$  or  $\pi / 5 \cdot 2^6$ . The final choice depends on satisfying Eq. (5) and also on the linearized input-output loop transfer function. As we are interested in FM demodulation, the DPLL transfer function should be as flat as possible out to  $f_m$ . Examining these conditions leads to the choice  $G_1 = \pi / 5 \cdot 2^4$ .

Summarizing, the second order DPLL gains are

$$G_1 = \pi / 5 \cdot 2^4 \quad G_2 = \pi / 5 \cdot 2^7$$

#### 7. Design of the Third Order DPLL

A double integral filter is added to the previous filter of Eq. (12):

$$H(z) = g_1 + g_2 \frac{1}{1 - z^{-1}} + g_3 \left( \frac{1}{1 - z^{-1}} \right)^2 \quad (15)$$

yielding the third order DPLL. For sinusoidal modulation, the phase error amplitude is approximately given by

$$(e_k)_{\max} = \frac{(2\pi f_m / f_s)^2}{G_3} 2\pi \Delta f / f_s \quad (16)$$

where  $G_3$  = gain of the double integral path. Imposing the phase error restriction yields (using  $\Delta f = 600$  Hz and  $f_m = 200$  Hz)  $G_3 \geq 0.000132$ . We choose  $G_3 = 0.000154 = \pi / 5 \cdot 2^{12}$  and  $m = 4$ . The values of  $G_1$  and  $G_2$  are chosen from the spike response and linearized loop characteristic. The spike responses illustrated in Fig. (5) show that there exist several pairs of gains  $G_1, G_2$  for which the third order DPLL suppresses the input spike, both for constant and sinusoidal modulation. To narrow the choice, a spike lasting  $2 / (\text{IF bandwidth})$  seconds is introduced and the final choice is

$$G_1 = \pi / 5 \cdot 2^5 \quad G_2 = \pi / 5 \cdot 2^8 \quad G_3 = \pi / 5 \cdot 2^{12} \quad (17)$$

These gains produce a linearized characteristic which is flat within 0.5 dB to 200 Hz and has a 730 Hz bandwidth, thereby satisfying Eq. (5).

## 8. Conclusions

An all digital phase-locked loop has been presented and a design procedure for a first, second, and third order loop is developed. The procedure involves quantization effects and response to artificial input noise spikes. Experimental results based on this design will be presented.

## References

1. An All Digital Phase-Locked Loop for FM Demodulation, J. Greco, J. Garodnick, D. L. Schilling International Conference on Communications, Phil. Pa., June 1972.

$$-2 \cos(\omega_0 t + \varphi(t)) + n(t)$$

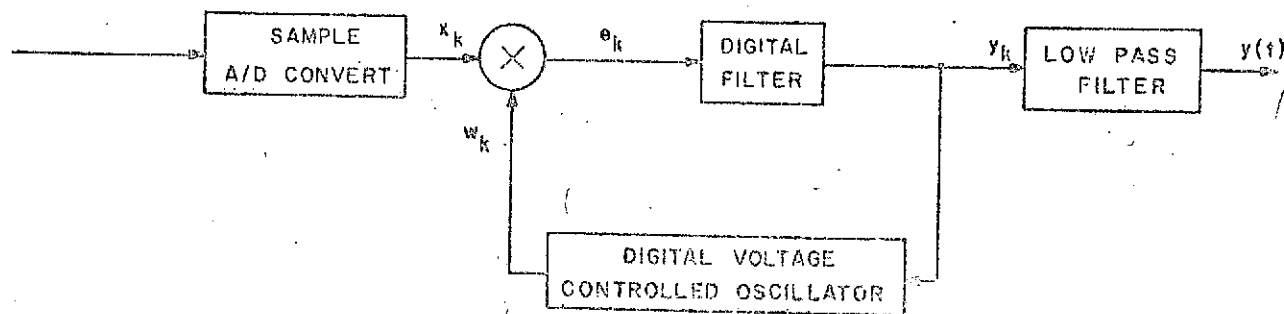


Fig. 1 Block diagram of the all digital phase-locked loop

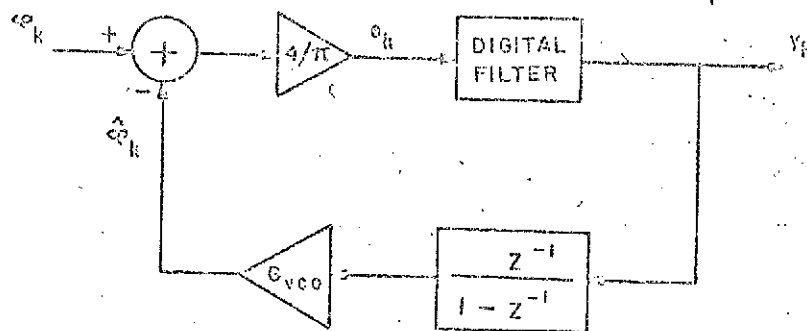


Fig. 2 Block diagram of the linearized model of the digital phase-locked loop

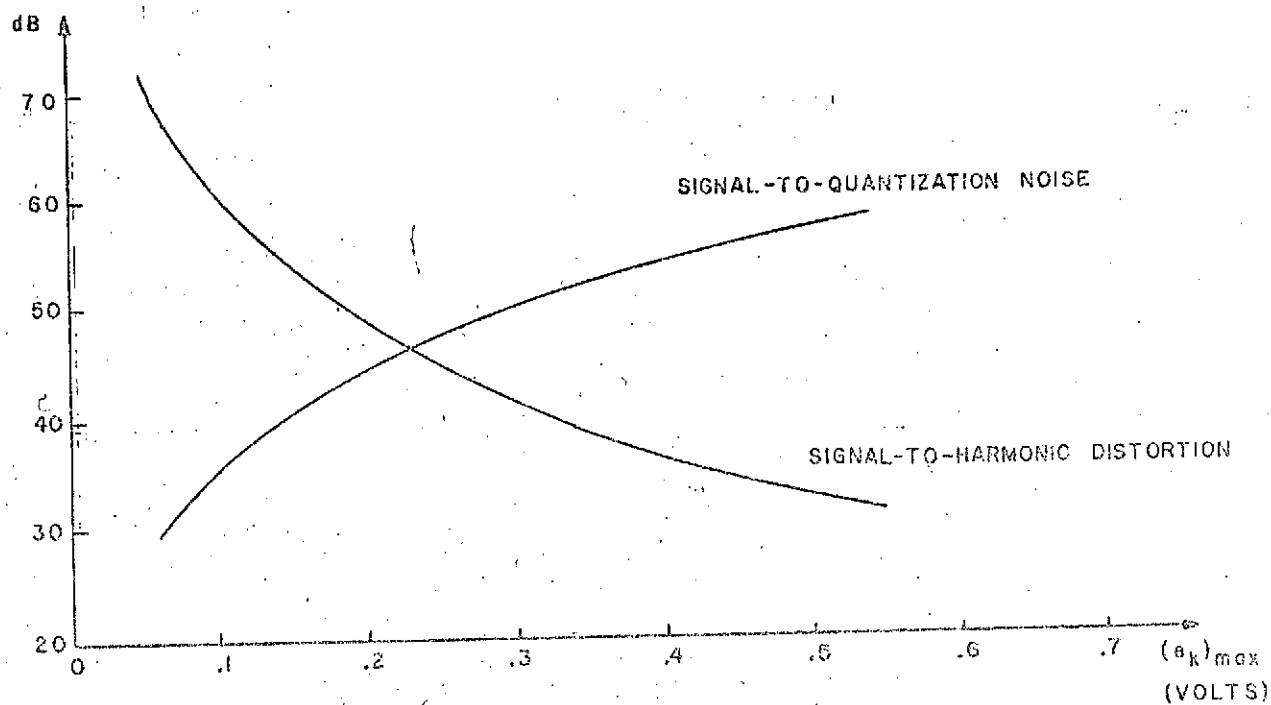


Fig. 3a Signal-to-quantization noise and signal-to-harmonic distortion as a function of maximum phase error for a 12-bit A/D converter.

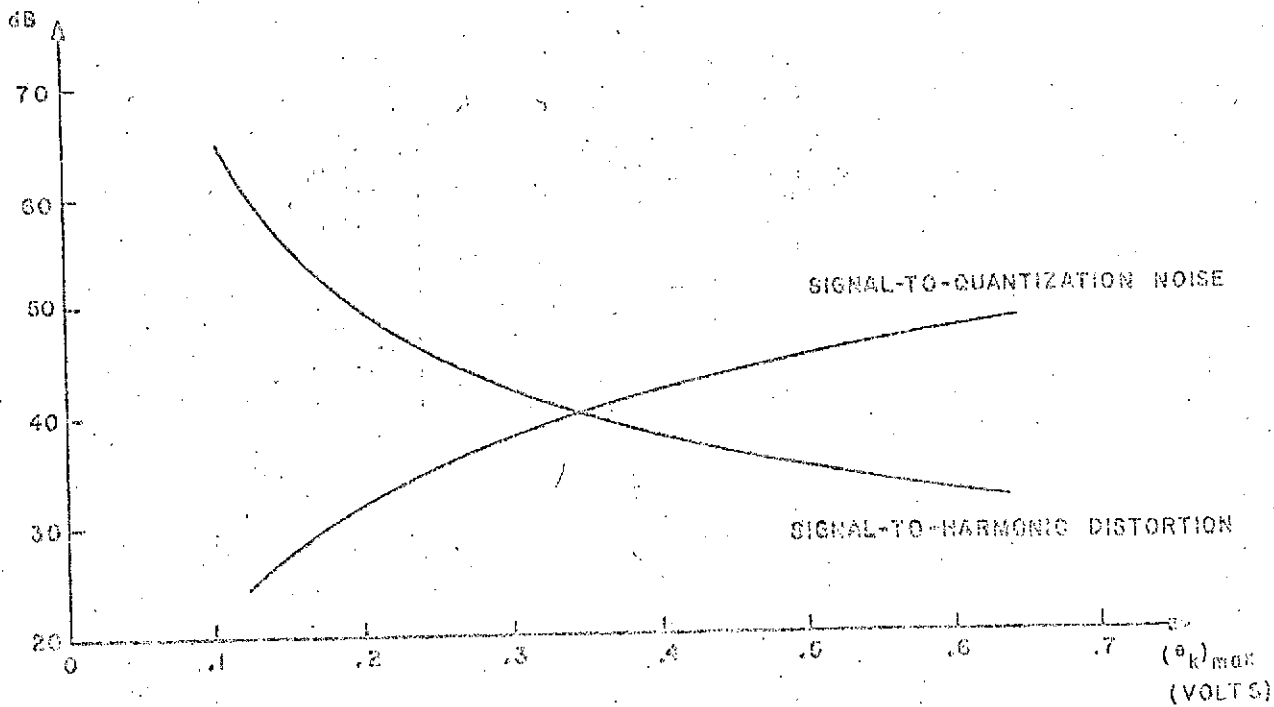
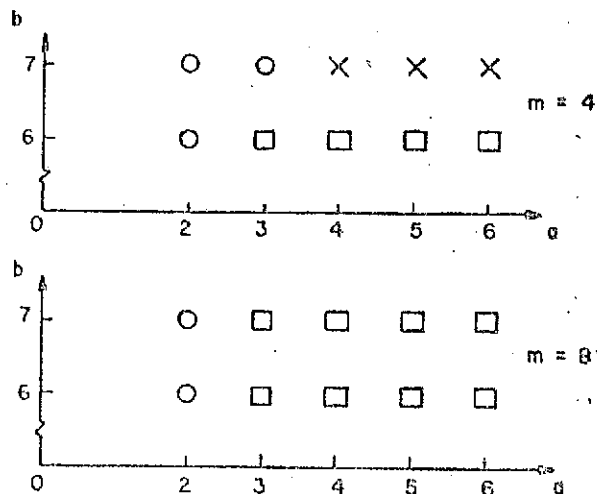


Fig. 3b Signal-to-quantization noise and signal-to-harmonic distortion as a function of maximum phase error for a 10-bit A/D converter.



$$G_1 = \pi/5 \cdot 2^a \quad G_2 = \pi/5 \cdot 2^b$$

× LOOP DOES NOT FOLLOW INPUT SPIKE

○ LOOP FOLLOWS INPUT SPIKE

□ LOOP BECOMES UNSTABLE

Fig. 4a. Second order DPLL spike response; the spike is superimposed on a 600 Hz frequency offset.

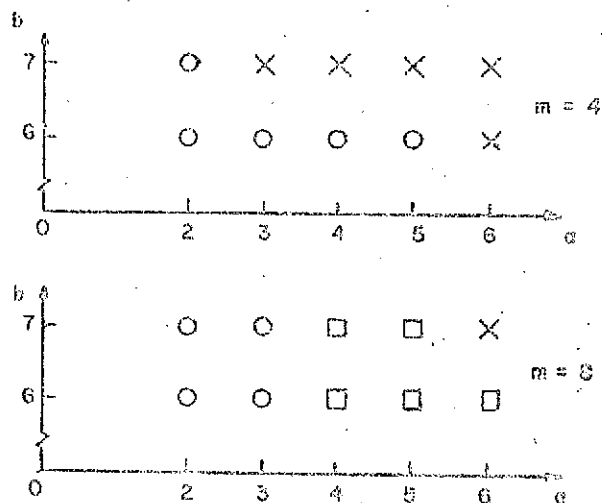
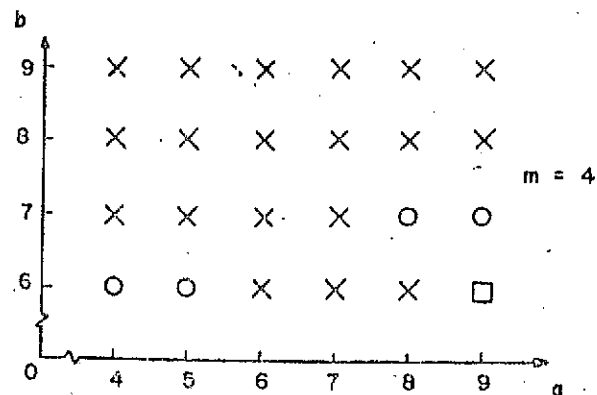
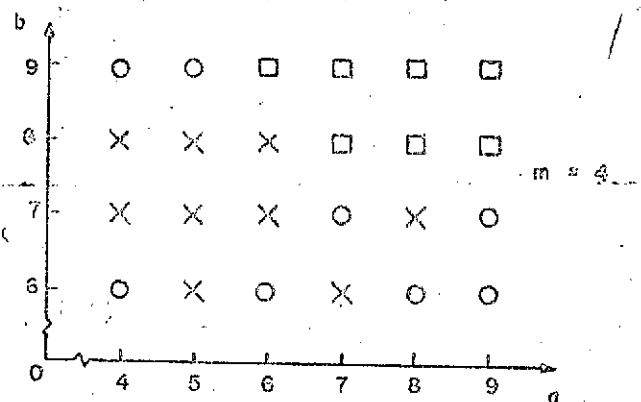


Fig. 4b. Second order DPLL spike response; the spike is superimposed on sinusoidal modulation with  $f_m = 200$  Hz and  $\beta = 3$ .



CONSTANT MODULATION  $\Delta f = 600$  Hz



SINUSOIDAL MODULATION  $\Delta f = 600$  Hz  $\beta = 3$

$$G_1 = \pi/5 \cdot 2^a, \quad G_2 = \pi/5 \cdot 2^b, \quad G_3 = \pi/5 \cdot 2^{12}$$

Fig. 5. Third order DPLL spike response.

# OVERSHOOT SUPPRESSION IN ADAPTIVE DELTA MODULATOR LINKS FOR VIDEO TRANSMISSION

by

L. Weiss, I. Paz, D. L. Schilling  
Department of Electrical Engineering  
The City College of C.U.N.Y.

## ABSTRACT

An overshoot suppression scheme to improve the performance of the Digital Song Adaptive Delta Modulator for picture transmission is described. The overshoot suppression algorithm has been verified using computer simulation on a PDP-8. It is also shown that the additional hardware required for the actual implementation of the algorithm is simpler than those encountered in the literature, and gives better signal tracking accuracy.

## I. INTRODUCTION

Delta Modulation is the name of the encoding process in a Digital Communications Link that allows for only changes in the input signal to be processed and transmitted. Using this procedure the link is usually required to transmit only one bit per sample, resulting in an improved bandwidth utilization efficiency over that obtained with PCM systems.

Briefly, a delta modulator operates as follows. The amplitude of the signal to be transmitted is sampled periodically and compared to an estimated value. The estimated signal is obtained by incrementing the previous estimate at each sampling time by a discrete amount called a step size. The sign of the difference between the signal and its estimate is used to decide if the previous estimate should be increased or decreased. This sign information (one bit per sample) is also transmitted over the channel to the decoder. The decoder uses this bit stream to construct the signal estimate.

A video signal is characterized by discontinuities of large amplitude and very short rise time. This corresponds to abrupt changes in shades in the picture content. A linear delta modulator is limited in its ability to track sudden input changes by its fixed step size. The magnitude of the steps is bounded by the permissible granular noise in constant shade regions, see Fig 1a. Shade contrast is thus degraded by the so called "slope-overload-noise" introduced by the delta modulators inability to rapidly follow the signal discontinuity. To alleviate this condition, while maintaining the permissible granular noise level, it is desirable to make the step sizes small initially but allow them to increase quickly in some nonlinear fashion when tracking a rapidly varying input, see Fig 1b. This is done in an adaptive delta modulator<sup>1,2</sup>.

The sharp rises in a video signal are usually followed by regions of constant level due to regions of uniform shade in the picture. Thus while alleviating slope overload problems, an adaptive delta modulator introduces the possibility of large overshoots when the tracked level is finally reached. Furthermore, the overshoot is followed by a transient oscillatory response until the delta modulator finally locks onto the tracked signal level.

These effects are shown in Fig 1(b).

Both the overshoot and the subsequent recovery time are undesirable attributes of an adaptive delta modulator. Reducing the step sizes decrease the possible overshoot amplitudes and shortens the recovery time. This, however, augments slope overload. A trade-off therefore exists between overshoot amplitude and recovery time versus slope-overload-noise in an adaptive delta modulator.

Overshoot suppression is a scheme to sharply limit the overshoot amplitude and reduce the subsequent recovery time. This is done without reducing the step sizes until overshoot is imminent. The trade-off between overshoot amplitude and recovery time versus slope-overload-noise is thus relaxed.

An overshoot suppression scheme has recently been suggested in the literature<sup>3</sup>. This scheme, however, uses a look-up table in which arbitrary values for the step sizes are suggested. It could therefore not serve the optimal delta modulator we are investigating in which the step sizes are obtained by explicit mathematical expressions. Furthermore, the maximum step size is limited in the above scheme by overshoot considerations. However, in the Song Delta Modulator, the step sizes can continually increase with the overshoot suppression scheme described below, thus yielding better signal to slope-overload-noise ratio. Moreover, the amount of equipment involved in implementing the proposed overshoot suppression algorithm is very modest in comparison to the equipment needed to implement other proposed schemes, and could fit into any adaptive delta modulator in which the next step size is explicitly calculated. It is also flexible as to the amount of overshoot suppression it can perform and trade-offs between conflicting factors can be accurately set as may be necessary.

## II. VIDEO TRANSMISSION CHARACTERISTICS OF THE DIGITAL-SONG-VARIABLE-STEP-SIZE DELTA MODULATOR

Figure 2 shows the structure of the digitally implemented optimum adaptive delta modulation system referred to in this paper. Briefly, its operation is as follows:

The input signal  $S(t)$  is sampled and A to D converted to give  $S_k$ .  $S_k$  is then compared to its estimate,  $X_k$ , generating a sign-bit  $e_k$ , with

$$e_k = \text{sgn.}(S_k - X_k) \quad (1)$$

where

$$X_k = X_{k-1} + \Delta_k \quad (2)$$

The step size at the  $k^{\text{th}}$  sampling instant is

$$\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1}) \quad (3)$$

Thus the  $k^{\text{th}}$  step size depends on the previous step size, and the previous two sign bits. The  $g_1$  and  $g_2$  function characteristics are shown in Fig 3 indicating that:

The research presented in this paper was partially supported by NASA grants NGR 33-013-063 and NGR 33-013-077.

$$\Delta_k = \begin{cases} |\Delta_{k-1}|(\alpha e_{k-1} + \beta e_{k-2}), & |\Delta_{k-1}| \geq 2\Delta_0 \\ 2\Delta_0 e_{k-1}, & |\Delta_{k-1}| < 2\Delta_0 \end{cases} \quad (4)$$

where  $\Delta_0$  is the minimum possible step size. The special region  $|\Delta_{k-1}| < 2\Delta_0$ , is needed to prevent a dead zone at the origin. The decoder is just the feedback portion of the encoder. It reconstructs the approximation  $X_k$  from the  $e_k$  sign bit stream.  $X_k$  is then D/A converted and low pass filtered to give  $\hat{S}(t)$ , the estimate of the transmitted signal.

In video processing,  $S(t)$  will contain many large discontinuities of very short rise time followed sometimes by constant levels. Thus edge response is extremely important in video. To permit  $\hat{S}(t)$  to approximate rapid rises, i.e., minimize slope-overload-noise, rapidly increasing step sizes are required. This can be accomplished by increasing  $\alpha$ ,  $\beta$ , as well as  $\Delta_0$  in (4). Gains in reducing slope-overload-noise are made at the expense of large overshoots and long subsequent recovery times. Furthermore, it can be seen in Fig 1(b) that good steady state response, i.e., small amplitude oscillations about a constant level in  $S(t)$ , requires small  $\Delta_0$ . It can also be shown that the delta modulator becomes unstable if  $\alpha$  and  $\beta$  are made too large. Thus in choosing  $\alpha$ ,  $\beta$ , and  $\Delta_0$  a trade-off must be made between slope overload noise versus overshoots, recovery time, and steady state response while maintaining delta modulator stability.

The addition of overshoot suppression to the adaptive delta modulator permits  $\alpha$  and  $\beta$  to be increased while decreasing  $\Delta_0$ . In this way, slope overload as well as small steady state response requirements can be met simultaneously. Impending instabilities due to large  $\alpha$  and  $\beta$  are also inhibited and, obviously, overshoots and subsequent recovery times are minimized.

### III. THE PROPOSED OVERSHOOT SUPPRESSION ALGORITHM

The Overshoot Suppression Algorithm may be understood by considering the four cases shown in Fig 4 in which an overshoot or an undershoot occurs. In Fig 4(a) an overshoot occurs at sampling time  $k-1$  followed immediately by an undershoot at  $k$ . For this case it is easy to show that the delta modulator considered will approach its steady-state condition rapidly. This is not the case in Fig 4(b) where the overshoot is larger than in (a) and  $X_k$  is greater than  $S_k$ . Consequently an undershoot occurs at  $k+1$  or later and with an amplitude larger than in (a). This occurs because the step sizes begin increasing again after the first reversed step. Thus it will take many more sampling periods to reach steady state in (b) than in (a). The algorithm is therefore implemented only when case (b) occurs. Note that Fig 4(c) and 4(d) depict undershoots corresponding to the overshoots in 4(a) and 4(b) respectively. Action to prevent excessive undershoots is thus taken only for case (d).

The occurrence of cases (b) and (d) can be recognized by examining the sequence  $(e_{k-2}, e_{k-1}, e_k)$ . The fingerprint of (b) is  $(+1, +1, -1)$ , while that of (d) is  $(-1, -1, +1)$ . When either sequence is encountered action is taken to prevent overshoot or undershoot.

The corrective action entails decreasing the stored values

of  $\Delta_{k-1}$  and hence  $X_{k-1}$  and  $X_k$ . Case (b) is thus transformed into a case (a) situation and the same for (d) and (c) respectively. The shape of the modified waveform actually depends on the amount by which  $\Delta_{k-1}$  is decreased. The simplest scheme is one where  $\Delta_{k-1}$  is replaced by half of its original value. We may allow for more rapidly increasing step sizes,  $\Delta_k$ , i.e., larger  $\alpha$  and  $\beta$  (see Eq 4), as long as  $\Delta_{k-1}$  is replaced by a smaller fraction of its original value when overshoot suppression is employed. Thus there is a faster initial rise coupled with a very sharp braking action just before the desired level is reached. However, since the braking action occurs close to the desired level there is nominal slope overload degradation. Indeed, there is an overall decrease in slope overload noise due to the more rapid increase in the initial step sizes.

Now the Overshoot Suppression Algorithm is applied to the adaptive delta modulator operating in the Song Mode, i.e.,  $\alpha=1$ ,  $\beta=0.5$ . It is shown elsewhere<sup>4</sup> that usable video transmission can be obtained using these parameters even without overshoot suppression. With the addition of the suppression algorithm video reproduction should be much improved.

The salient features of the Song Mode response are now summarized. In approaching a level from above or below as in Fig 4, each step size is 1.5 times the previous one (see Eq 4 for  $\alpha=1$ ,  $\beta=0.5$ ). When a direction reversal occurs, as at sampling time  $k$  in Fig 4, then the first step size following the reversal is one half the previous step size, i.e.,  $\Delta_k = \frac{1}{2} \Delta_{k-1}$  (see Eq 4). Thus, in Fig 4(b) we have

$$X_k \geq X_{k-1} - \frac{1}{2} \Delta_{k-1} = X_{k-2} + \frac{1}{2} \Delta_{k-1} \quad (5)$$

The inequality sign is needed due to the fixed point arithmetic employed in the digital implementation. Also in Fig 4(b)  $X_{k-2} < S_k < X_k$ . To implement overshoot suppression set  $(\Delta_k)' = \frac{1}{2} \Delta_{k-1}$ , where the prime refers to the new values after the overshoot suppression algorithm has been implemented. Therefore,  $(X_k)' = X_{k-2} + (\Delta_k)' = X_{k-2} + \frac{1}{2} \Delta_{k-1}$  (6)

Next, set

$$(\Delta_k)' = \Delta_k = -\frac{1}{2} \Delta_{k-1} \quad (7)$$

Thus

$$(X_k)' = X_{k-1}' + (\Delta_k)' = X_{k-2} \quad (8)$$

Hence, Fig 4(b) has been transformed into Fig 5, with undershoot occurring at  $k$  rather than  $k+1$  or later. It should be evident from Fig 5 even without a detailed explanation of the worst case that the overshoot has been at best entirely eliminated or at worst cut in half depending on whether  $S(t)$  is closer to  $(X_{k-1})'$  or  $X_{k-2}$  respectively. Figure 5 also shows that the recovery time is greatly reduced, since the delta modulator locks onto  $S(t)$  very rapidly after sampling time  $k$ . Note also that now  $(e_k)' = \text{sgn.}(S_k - (X_k)') = +1$ , whereas in Fig 4(b)  $e_k = \text{sgn.}(S_k - X_k) = -1$ .

The above overshoot suppression scheme is now summarized in the form of an algorithm by considering a typical cycle of the now modified delta modulator.

Step 1: Generate  $S_k$   
 Step 2: Calculate  $\hat{X}_k = g_1(e_{k-1}, \hat{A}_{k-1}) + g_2(e_{k-2}, \hat{A}_{k-1})$   
 Step 3: Calculate  $X_k = X_{k-1} + \hat{A}_k$   
 Step 4: Calculate  $e_k = \text{sgn}(S_k - X_k)$  and transmit this bit.

In the delta modulator without overshoot suppression this would complete the cycle. That is,  $k$  is next updated and steps through 4 are repeated. To implement overshoot suppression the following additional steps are needed.

Step 5: If  $e_{k-3} = e_{k-2} = +1$ , and  $e_{k-1} = e_k = -1$ , set  $V=1$ .  
 If  $e_{k-3} = e_{k-2} = -1$ , and  $e_{k-1} = e_k = +1$ , set  $W=1$ .  
 Step 6: If  $V \neq 1$  and  $W \neq 1$  go to 7, otherwise set  
 (a)  $(\hat{A}_{k-1})' = \frac{1}{2} \hat{A}_{k-1}$   
 (b)  $(X_{k-1})' = X_{k-2} + (\hat{A}_{k-1})' = X_{k-2} + \frac{1}{2} \hat{A}_{k-1}$   
 (c)  $(\hat{A}_k)' = \hat{A}_k = -\frac{1}{2} \hat{A}_{k-1}$   
 (d)  $(X_k)' = (X_{k-1})' + (\hat{A}_k)' = X_{k-2}$   
 (e)  $(e_k)' = -e_k$   
 Step 7: Update  $k$ . That is, set  $e_{k-2} = e_{k-1}$ ,  $e_{k-1} = e_k$ ; if step 6 is executed  $e_{k-2} = (e_k)'$ , otherwise  $e_{k-1} = e_k$ , etc.

#### IV. HARDWARE IMPLEMENTATION OF THE OVERSHOOT SUPPRESSION ALGORITHM

The implementation of the above overshoot suppression algorithm requires the addition of very little hardware to the Digital Song Adaptive Delta Modulator. This can be seen by considering the schematic representation of the delta modulator CODEC (Coder-Decoder-Combination) with overshoot suppression shown in Fig. 2. Note that the extra components needed to implement the suppression scheme appear in branches that are drawn with dashed lines. Of these, the only major devices are the delay elements D5, D6, and the adder A4. However, since the adders A1, A2, and A3 are really one time-shared adder, we can easily time-share adder A4 also. The remaining extra elements are only a few gates needed for decision, switching, and timing purposes. Note that the execution of step 6(a) of the algorithm,  $(\hat{A}_{k-1})' = \frac{1}{2} \hat{A}_{k-1}$ , need not require the addition of any explicit hardware. We merely read into adder A4 the contents of the  $\hat{A}_{k-1}$  register shifted by one bit, thereby resulting in a division by two.

It is difficult to discern the operation of the circuit by merely examining the schematic diagram in Fig. 2 because the sequential order of operations is not specified in the diagram. However, the actual operation is made clear by considering Fig. 2 in conjunction with the seven steps of the overshoot suppression algorithm.

The additional steps of the algorithm place an added requirement on the logic speed. After the completion of a normal cycle of the delta modulator, extra time is needed to perform two more additions and the various logic operations needed to rearrange the internal values. No problem will arise if this can be done in one sampling period. If it cannot, then either the sampling rate must be decreased, or logic components capable of higher switching speed must be used.

At this point, it should be pointed out that a significant

simplification is possible in the encoder implementation. Namely, Step 6(b) and hence Step 6(a) do not have to be explicitly executed in the encoder because  $(X_{k-1})'$  is not really needed to compute  $(X_k)'$  in Step 6(d), i.e.,  $(X_k)'$  is simply replaced by  $X_{k-2}$  which is available in the memory. Furthermore, it is easy to show that  $(X_{k-1})'$  is not used in later cycles due to the fact that once an overshoot is suppressed at  $k-1$ , the earliest future time for implementing the algorithm is at  $k+2$ . By this time  $X_{k-1}$  is clocked out of the memory. In terms of hardware savings in the encoder, this eliminates the gates needed to produce  $(\hat{A}_{k-1})' = \frac{1}{2} \hat{A}_{k-1}$ , as well as A4 and A2 to carry out Steps 6(b) and 6(d) respectively. These simplifications are not possible in the decoder because its output with overshoot suppression, has to be taken from  $X_{k-1}$  rather than from  $X_k$ . Note that if the output was taken from  $X_k$ , then the overshoot suppression produced by going back in time and reducing  $X_{k-1}$  would not be evident in the output  $\hat{S}(t)$ .

#### V. COMPUTER SIMULATIONS

The Digital Song Delta Modulator, with and without overshoot suppression, was simulated on a PDP-8 computer. The minimum step size used ( $\Delta_0$ ) was normalized to unity. The dynamic range was 0 to 1024  $\Delta_0$ . This corresponds to a ten bit internal arithmetic in an actual hardware implementation.

The responses of the delta modulator to step functions of different amplitudes, with and without overshoot suppression, appear in Fig. 6. Figs. 6(a), and 6(b) exhibit large overshoots and sustained oscillations. They correspond to the sequence  $e_{k-3} = e_{k-2} = +1$ ,  $e_{k-1} = e_k = -1$ , where  $k-1$  is the sampling time when overshoot occurs. Figs. 6(a'), and 6(b') are the same waveforms but with overshoot suppression. As an example, compare Figs. 6(a) and 6(a'). Here the maximum peak-to-peak oscillations are reduced from  $22\Delta_0$  to  $9\Delta_0$ . Similar observations can be made for Figs. 6(b) and 6(b'). Furthermore, here the settling time to the steady state is reduced from six to three sampling periods.

While Fig. 6 gives a good indication of the general nature of the improvement, due to overshoot suppression, a more convincing illustration is depicted in Fig. 7 where the discontinuities are much larger. Note that the apparent slow rise times in Fig. 7 are due to the compression produced by a scaling factor of 0.1 used in the plotting. In reality Fig. 7 rises over a range of  $5000\Delta_0$  in only 13 sampling periods. To achieve the same amplitude, a non-adaptive delta modulator would require 500 sampling periods.

Briefly, the salient features of the response are as follows: The rise time to reach a given level is the same with or without overshoot suppression. Overshoots are suppressed by a minimum of 50%. Recovery times following overshoots are significantly reduced as seen in Fig. 7b. The data plotted in Fig. 7 is given in Table 1 for quantitative comparisons. The peak-to-peak amplitude of the steady state response is three times the minimum step size for either scheme. The period of steady state oscillations is 4 sampling periods without overshoot suppression, and 8 sampling periods with overshoot suppression. In either case, the peak-to-peak steady state oscillation amplitudes are smaller than a grey level in the picture waveform. Thus constant shade regions will not suffer significant degradation.

This is based on the assumption of quantizing a pixel into 64 grey levels with a total dynamic range of 1024Δ

## VI. CONCLUSIONS

An overshoot suppression algorithm has been proposed and verified by computer simulation. It has been shown that the scheme significantly improves the transient behavior of video waveforms transmitted using delta modulation techniques.

The main advantages of the proposed algorithm are:

- It can be easily utilized in optimal digital delta modulators that can be described by a closed-form mathematical formulation and in particular in the Adaptive Song Delta Modulator.
- The scheme has rather modest requirements for hardware implementation.
- It allows for flexible trade-off between slope-overload and overshoot noise.

Therefore, the addition of the overshoot suppression algorithm significantly improves the performance of the digital delta modulator for picture transmission.

## VII. REFERENCES

- C.L. Song, J. Garodnick, and D.L. Schilling, "A Variable-Step-Size Robust Delta Modulator," IEEE Trans. Commun. Technol., Vol COM-19, pp. 1033-1044, Dec. 1971.
- M.R. Winkler, "Pictorial Transmission with HIDM" in 1965 IEEE Int. Conv. Rec., pt. 1, pp.285-291.
- M. Oliver, "An Adaptive Delta Modulator with Overshoot Suppression for Video Signals," IEEE Trans. Commun. Technol., Vol. COM-21, pp. 243-247, March 1973.
- C.L. Song, J. Garodnick, D.L. Schilling, "An Adaptive Delta Modulator for Speech and Video Processing," Proceedings of the 1972 IEEE International Conference on Commun., Philadelphia, Pa., pp.21.30-21.31, June, 1972.

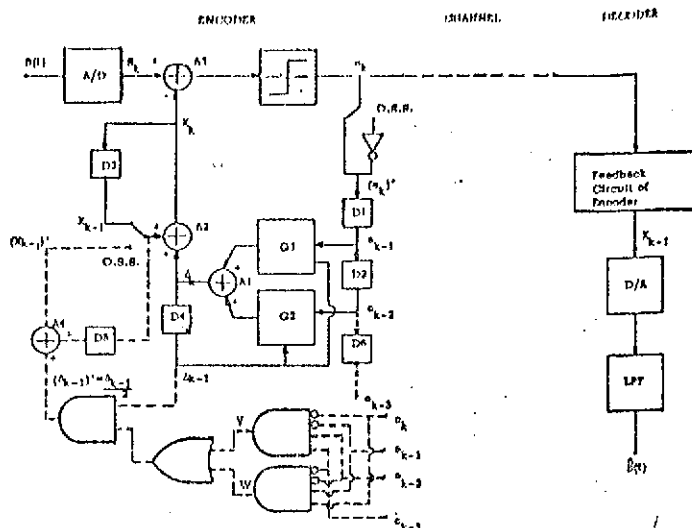
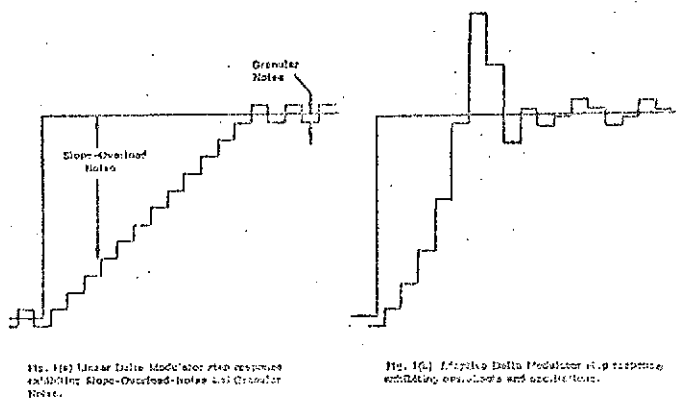


Fig. 2 Song-Adaptive-Delta Modulator with provisions for overshoot suppression. (Dashed branches are for overshoot suppression.)

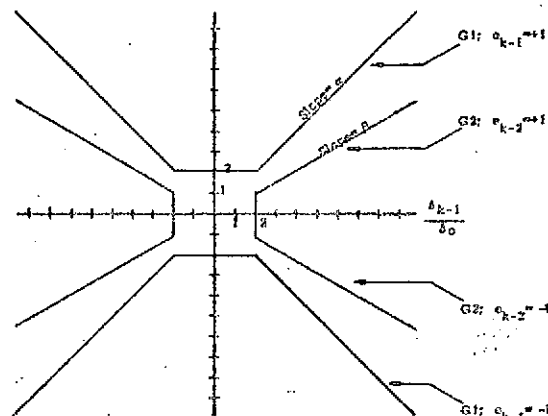


Fig. 3 Normalized  $G_1, G_2$  characterization. ( $\Delta_c = \text{Minimum Step Size}$ )

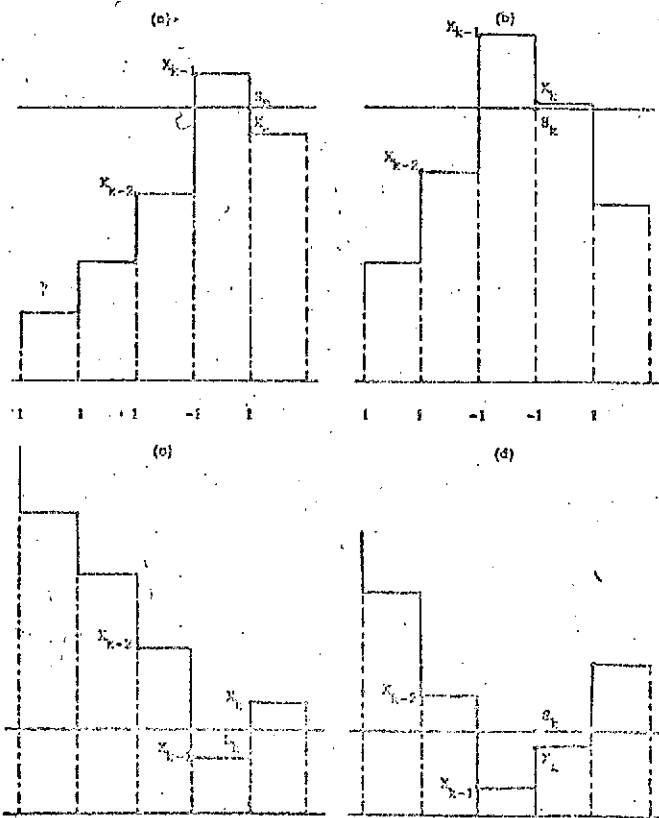


Fig. 4 The four cases of overshoot in the step response of the Song-Adaptive-Delta Modulator.

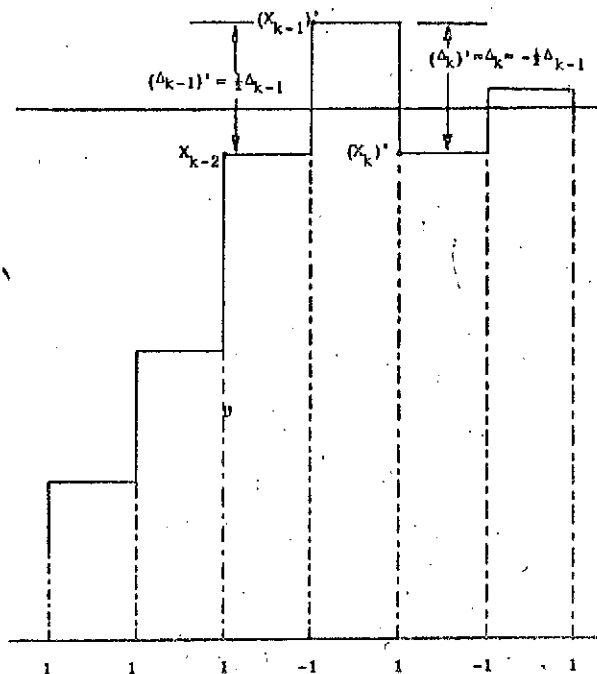


Fig. 5. Figure 4b after Overshoot Suppression.

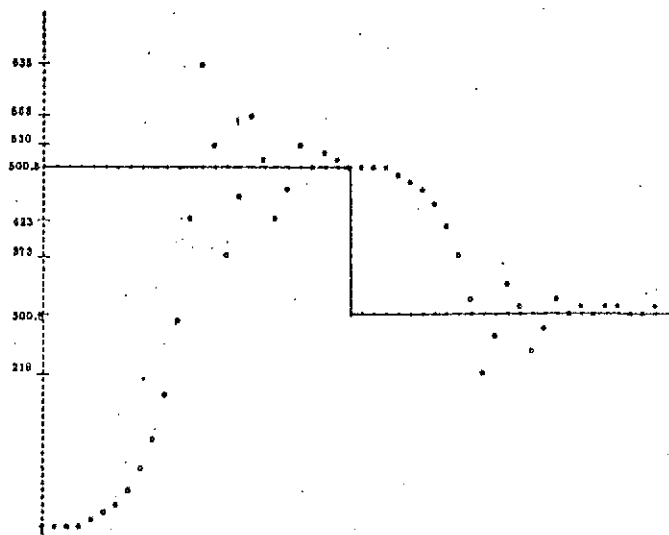


Fig. 7(a). The response of a Long-Adaptive Delta Modulator to abrupt level changes without Overshoot Suppression.

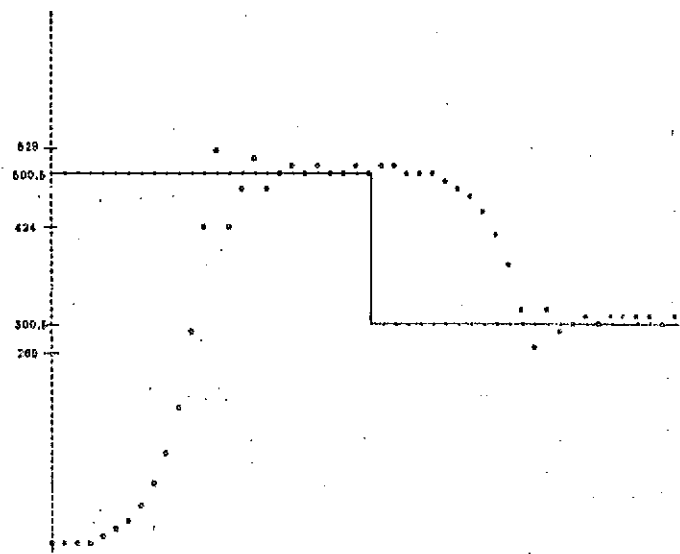


Fig. 7(b). The response of a Long-Adaptive Delta Modulator to abrupt level changes with Overshoot Suppression.

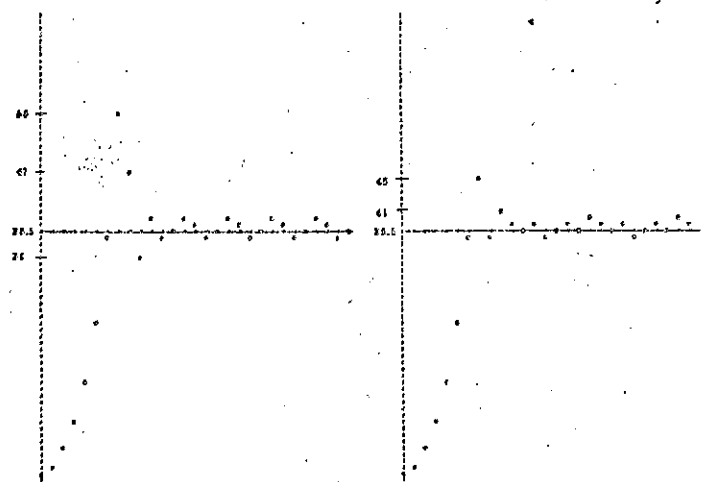


Fig. 8(a). The step response of a Long-Adaptive Delta Modulator without Overshoot Suppression.

Fig. 8(b). The step response of a Long-Adaptive Delta Modulator with Overshoot Suppression.

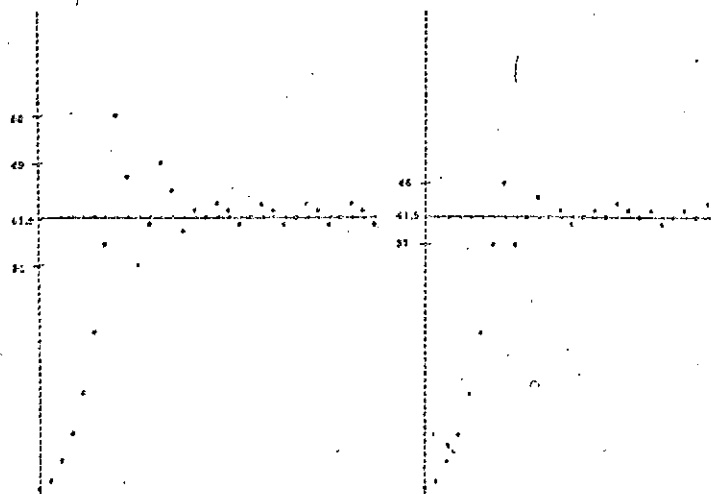


Fig. 8(b). The step response of a Long-Adaptive Delta Modulator without Overshoot Suppression.

Fig. 8(c). The step response of a Long-Adaptive Delta Modulator with Overshoot Suppression.

K= 0.000	SK= 500.5	XK= 0.000	EKE= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKE= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKE= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKE= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKE= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKE= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKE= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKE= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKE= 1.000
K= 9.000	SK= 500.5	XK= 126.00	EKE= 1.000
K= 10.00	SK= 500.5	XK= 169.00	EKE= 1.000
K= 11.00	SK= 500.5	XK= 233.00	EKE= 1.000
K= 12.00	SK= 500.5	XK= 324.00	EKE= 1.000
K= 13.00	SK= 500.5	XK= 455.00	EKE= 1.000
K= 14.00	SK= 500.5	XK= 630.00	EKE= 1.000
K= 15.00	SK= 500.5	XK= 863.00	EKE= 1.000
K= 16.00	SK= 500.5	XK= 1180.00	EKE= 1.000
K= 17.00	SK= 500.5	XK= 1600.00	EKE= 1.000
K= 18.00	SK= 500.5	XK= 2150.00	EKE= 1.000
K= 19.00	SK= 500.5	XK= 2960.00	EKE= 1.000
K= 20.00	SK= 500.5	XK= 4060.00	EKE= 1.000
K= 21.00	SK= 500.5	XK= 5500.00	EKE= 1.000
K= 22.00	SK= 500.5	XK= 7420.00	EKE= 1.000
K= 23.00	SK= 500.5	XK= 10000.00	EKE= 1.000
K= 24.00	SK= 500.5	XK= 13400.00	EKE= 1.000
K= 25.00	SK= 500.5	XK= 18000.00	EKE= 1.000
K= 26.00	SK= 300.5	XK= 24200.00	EKE= 1.000
K= 27.00	SK= 300.5	XK= 32600.00	EKE= 1.000
K= 28.00	SK= 300.5	XK= 43800.00	EKE= 1.000
K= 29.00	SK= 300.5	XK= 58400.00	EKE= 1.000
K= 30.00	SK= 300.5	XK= 77000.00	EKE= 1.000
K= 31.00	SK= 300.5	XK= 101000.00	EKE= 1.000
K= 32.00	SK= 300.5	XK= 132000.00	EKE= 1.000
K= 33.00	SK= 300.5	XK= 171000.00	EKE= 1.000
K= 34.00	SK= 300.5	XK= 219000.00	EKE= 1.000
K= 35.00	SK= 300.5	XK= 286000.00	EKE= 1.000
K= 36.00	SK= 300.5	XK= 374000.00	EKE= 1.000
K= 37.00	SK= 300.5	XK= 496000.00	EKE= 1.000
K= 38.00	SK= 300.5	XK= 656000.00	EKE= 1.000
K= 39.00	SK= 300.5	XK= 868000.00	EKE= 1.000
K= 40.00	SK= 300.5	XK= 1136000.00	EKE= 1.000
K= 41.00	SK= 300.5	XK= 1574000.00	EKE= 1.000
K= 42.00	SK= 300.5	XK= 2090000.00	EKE= 1.000
K= 43.00	SK= 300.5	XK= 2800000.00	EKE= 1.000
K= 44.00	SK= 300.5	XK= 3740000.00	EKE= 1.000
K= 45.00	SK= 300.5	XK= 4960000.00	EKE= 1.000
K= 46.00	SK= 300.5	XK= 6560000.00	EKE= 1.000
K= 47.00	SK= 300.5	XK= 8680000.00	EKE= 1.000
K= 48.00	SK= 300.5	XK= 11360000.00	EKE= 1.000
K= 49.00	SK= 300.5	XK= 15740000.00	EKE= 1.000
K= 50.00	SK= 300.5	XK= 20900000.00	EKE= 1.000

Table 1(a) Table of values for Fig. 7(a)

K= K'th sampling instant.

SK = input signal at time K.

XK = input estimate at time K.

EKE = sign bit at time K.

K= 0.000	SK= 500.5	XK= 0.000	EKU= 1.000	EKT= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKU= 1.000	EKT= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKU= 1.000	EKT= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKU= 1.000	EKT= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKU= 1.000	EKT= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKU= 1.000	EKT= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKU= 1.000	EKT= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKU= 1.000	EKT= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKU= 1.000	EKT= 1.000
K= 9.000	SK= 500.5	XK= 126.00	EKU= 1.000	EKT= 1.000
K= 10.00	SK= 500.5	XK= 169.00	EKU= 1.000	EKT= 1.000
K= 11.00	SK= 500.5	XK= 233.00	EKU= 1.000	EKT= 1.000
K= 12.00	SK= 500.5	XK= 324.00	EKU= 1.000	EKT= 1.000
K= 13.00	SK= 500.5	XK= 455.00	EKU= 1.000	EKT= 1.000
K= 14.00	SK= 500.5	XK= 630.00	EKU= 1.000	EKT= 1.000
K= 15.00	SK= 500.5	XK= 863.00	EKU= 1.000	EKT= 1.000
K= 16.00	SK= 500.5	XK= 1180.00	EKU= 1.000	EKT= 1.000
K= 17.00	SK= 500.5	XK= 1600.00	EKU= 1.000	EKT= 1.000
K= 18.00	SK= 500.5	XK= 2150.00	EKU= 1.000	EKT= 1.000
K= 19.00	SK= 500.5	XK= 2960.00	EKU= 1.000	EKT= 1.000
K= 20.00	SK= 500.5	XK= 4060.00	EKU= 1.000	EKT= 1.000
K= 21.00	SK= 500.5	XK= 5500.00	EKU= 1.000	EKT= 1.000
K= 22.00	SK= 500.5	XK= 7420.00	EKU= 1.000	EKT= 1.000
K= 23.00	SK= 500.5	XK= 10000.00	EKU= 1.000	EKT= 1.000
K= 24.00	SK= 500.5	XK= 13400.00	EKU= 1.000	EKT= 1.000
K= 25.00	SK= 500.5	XK= 18000.00	EKU= 1.000	EKT= 1.000
K= 26.00	SK= 300.5	XK= 24200.00	EKU= 1.000	EKT= 1.000
K= 27.00	SK= 300.5	XK= 32600.00	EKU= 1.000	EKT= 1.000
K= 28.00	SK= 300.5	XK= 43800.00	EKU= 1.000	EKT= 1.000
K= 29.00	SK= 300.5	XK= 58400.00	EKU= 1.000	EKT= 1.000
K= 30.00	SK= 300.5	XK= 77000.00	EKU= 1.000	EKT= 1.000
K= 31.00	SK= 300.5	XK= 101000.00	EKU= 1.000	EKT= 1.000
K= 32.00	SK= 300.5	XK= 132000.00	EKU= 1.000	EKT= 1.000
K= 33.00	SK= 300.5	XK= 171000.00	EKU= 1.000	EKT= 1.000
K= 34.00	SK= 300.5	XK= 219000.00	EKU= 1.000	EKT= 1.000
K= 35.00	SK= 300.5	XK= 286000.00	EKU= 1.000	EKT= 1.000
K= 36.00	SK= 300.5	XK= 374000.00	EKU= 1.000	EKT= 1.000
K= 37.00	SK= 300.5	XK= 496000.00	EKU= 1.000	EKT= 1.000
K= 38.00	SK= 300.5	XK= 656000.00	EKU= 1.000	EKT= 1.000
K= 39.00	SK= 300.5	XK= 868000.00	EKU= 1.000	EKT= 1.000
K= 40.00	SK= 300.5	XK= 1136000.00	EKU= 1.000	EKT= 1.000
K= 41.00	SK= 300.5	XK= 1574000.00	EKU= 1.000	EKT= 1.000
K= 42.00	SK= 300.5	XK= 2090000.00	EKU= 1.000	EKT= 1.000
K= 43.00	SK= 300.5	XK= 2800000.00	EKU= 1.000	EKT= 1.000
K= 44.00	SK= 300.5	XK= 3740000.00	EKU= 1.000	EKT= 1.000
K= 45.00	SK= 300.5	XK= 4960000.00	EKU= 1.000	EKT= 1.000
K= 46.00	SK= 300.5	XK= 6560000.00	EKU= 1.000	EKT= 1.000
K= 47.00	SK= 300.5	XK= 8680000.00	EKU= 1.000	EKT= 1.000
K= 48.00	SK= 300.5	XK= 11360000.00	EKU= 1.000	EKT= 1.000
K= 49.00	SK= 300.5	XK= 15740000.00	EKU= 1.000	EKT= 1.000

Table 1(b) Table of values for Fig. 7(b)

EKU = sign bit used inside encoder and decoder.

EKT = sign bit transmitted from encoder to decoder.